NAVIGATION, GUIDANCE AND CONTROL FOR PLANETARY LANDING

NAVIGATION, GUIDANCE AND CONTROL FOR PLANETARY LANDING

CARLOS NORBERTO PEREZ MONTENEGRO
POLITECNICO DI TORINO

TURIN, ITALY
2013

NAVIGATION, GUIDANCE AND CONTROL FOR PLANETARY LANDING

FINAL DISSERTATION

Ph.D. IN COMPUTER AND CONTROL ENGINEERING

POLITECNICO DI TORINO

TURIN, ITALY

CARLOS NORBERTO PEREZ MONTENEGRO

POLITECNICO DI TORINO

DIRECTED BY

ENRICO CANUTO

POLITECNICO DI TORINO, ITALY

TURIN, ITALY

2013

Notation

Vector	$\vec{v}$
Vector Representation	$\mathbf{v}$
Vector Components	$\mathbf{v} = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix}$
Vector in a local Reference Frame	$\mathbf{v}_l = \begin{bmatrix} v_{lx} \\ v_{ly} \\ v_{lz} \end{bmatrix}$

ACKNOWLEDGMENTS

Quiero agradecer Principalmente a Carolina Cardenas quien con su amor y apoyo incondicional me dio el valor y fortaleza para terminar este ciclo de estudio. A mis padres que siempre se encuentran conmigo sin importar la distancia me hicieron sentir siempre que se encuentran a mi lado apoyándome de corazón y espíritu, a mi hermano que siempre será mi modelo a seguir trabajando incansablemente para lograr sus metas aun en los momentos difíciles.

Agradezco a:
Mi tutor Enrico Canuto, quien ha sido un gran guía y en estos años ha sido como parte de mi familia.

A todos mis amigos y colegas que me apoyaron en este proyecto, Andres Molano, Jose Ospina, Mauricio Lotufo y Wilber Acuna quienes me ayudaron en muchos momentos difíciles de esta investigación.

A mis amigos Astrid Calderón, Adolfo Vargas, Antonela Olivo, Alejandra Rodríguez, Diomar Elena Calderon, Verónica Vergara, Madeleleyn Mendosa, Irene Carreño, Nicolás Salazar, Maria Paula Salazar, Ricardo Rugeles, Sandra Gómez, Sergio Ortiz, German Cortez, Marisol Vargas, Mirko Bocco, Antonio Lopera, Javier Martínez, Felipe Gonzales, Yineth Carolina Valero, Antonnino Scoma, Nataly Guataquira y Paula Caballero con los que compartí momentos muy especiales en este periodo de mi vida y estuvieron conmigo en cientos de fechas especiales e inolvidables.

A mi abuela Inés Castillo mi gran maestra.

A Osiris, Isis, Zeus, Nefertiti y Hercules que me acompañaron en la distancia hasta donde la vida les permitió, pero su recuerdo me ayuda a no perder nunca mi sonrisa.

Contents

PART I NAVIGATION, GUIDANCE AND CONTROL FOR PLANETARY LANDING

1. INTRODUCTION

This dissertation aims to develop algorithms of guidance and control for propulsive terminal phase planetary landing, including a piloting strategy. The algorithms developed here are based on the Embedded Model Control (EMC) principles [1]–[6]. This research treats an extension of the architecture proposed in Molano's PhD dissertation [7]. Currently, the planetary entry descent and landing are important issues, landing on Mars and Moon has been scientifically rewarding; successful landed robotic systems on the surface of Mars have been achieved. Projects as Mars Science Laboratory MSL [8]–[10] inter alia have achieved a successful landing. These new approaches are focused in delivering large amounts of mass with a low uncertainty and in performing the entry, descent and landing sequence for human exploration. This dissertation treats the last phase of the planetary landing with a pinpoint landing strategy [11], [12]. The dissertation is divided in two parts, the first part is focused on Pinpoint landing algorithms that have been studied in recently years[7], [12]–[17] and the integration between the guidance and the piloting. Chapter 2 describes the phases of the entry descent and landing (EDL) phases and the frames of references that are involve in the last phase of landing (propulsive phase). Then, a geometric description of the propulsive system and an optimal dispatching strategy for a generic case is depth on chapter 3. The guidance and control for planetary landing and the complete design follows the EMC methodology is described in Chapter 4, where a unique discrete-time state equation (the embedded model EM) is derived and used by the Guidance Navigation and Control (GNC). Here only guidance and piloting are treated. The whole GNC algorithm has been tested on a simulator. In chapter 5 a hazard avoidance strategy is developed based on computer vision process [18]–[20], piloting definition and its integration with guidance is studied and some simulations runs are provided.

On the other hand the development of this project allowed an alternative methodology to model and control a small quadrotor for testing propulsive planetary landing, guidance, navigation and control called project Borea [21]–[26]. The second part of this research describes this project. Chapter 7 shows modelling of quadrotor dynamics and kinematics. Its propulsive system is studied and an alternative methodology for the propeller modelling

is presented. The embedded model for quadrotor vehicles is developed in chapter 8. Vertical and horizontal position guidance is developed on chapter 9, a high level navigation is described on chapter 10 and the control law is explained on chapter 11. The problem of on-ground testing guidance, navigation and control (GNC) algorithms for accurate and safe planetary landing can be approached through the flight of small quadrotors, suitable for indoor and outdoor operations. The dissertation is focused on the test of GNC algorithms for planetary landing. The main difference of an on-Earth-flying quadrotor dynamics with respect to a generic planetary landing vehicle is analyzed in chapter 12.

2. PLANETARY DESCENT DYNAMICS

2.1 Entry, Descent and Landing

The planetary landing can be concentrated in the main four phases covered in these summary [20], [27], [28]; they are mentioned and explained below and also shown in Figure 1;

1) Approach phase
2) Entry phase
3) Parachute phase
4) Powered descent phase

In the approach phase trajectory rectification manoeuvres are executed and, the navigation is performed on the ground using radiometric tracking data, the predicted position, velocity and attitude [7], [10], [29]. Next in the entry phase starts at entry interface, during this phase the entry controller achieve the commanded 3-axis attitude by generating roll, pitch, and yaw torque commands, The purposes of this phase are to survive the entry environment, as well as aeroheating heat pulse and to reach the desired parachute deploy target. The latter objective is approached by the so called guided entry: the vehicle is endowed with a (small) lift force and the lift direction in the vertical plane is suitably oriented by thrusters. Guide entry was employed by the US shuttles when entering in the Earth atmosphere and for the first time outside the Earth by the US Mars Science Laboratory in 2012 [8]–[10], [27]. Guided entry allowed MSL to reduce landing uncertainty to well below 10 km. An approach to guided entry with EMC is in [1], [2], [6]. The parachute phase starts at altitudes of about 10 km above Mars' surface. On Mars due to low atmosphere density, parachute landing is not possible, and supersonic parachute must be deployed, contrary to the Earth entry. Parachute descent allows the vehicle sped to be reduced from about 500 m/s to less than 100 m/s, when the propulsive phase starts. In this phase the spacecraft is reconfigured with the jettisoning of the heat-shield, which exposes the local sensors, allowing the vehicle's altitude and velocity measurements. During parachute phase the vehicle trajectory (especially the horizontal component) is exposed to winds that may generate unwanted displacements up to 4 km. Pinpoint landing thus

requires to recover such displacement by controlling the lateral vehicle position. Up to now nothing of this sort has been done on Mars. Also MSL was missing lateral control except for diverting the vehicle from the parachute and shield after their jettisoning. This dissertation outlines a guidance that allows also horizontal control of the vehicle. The last phase, powered descent phase is explained in section 2.2.

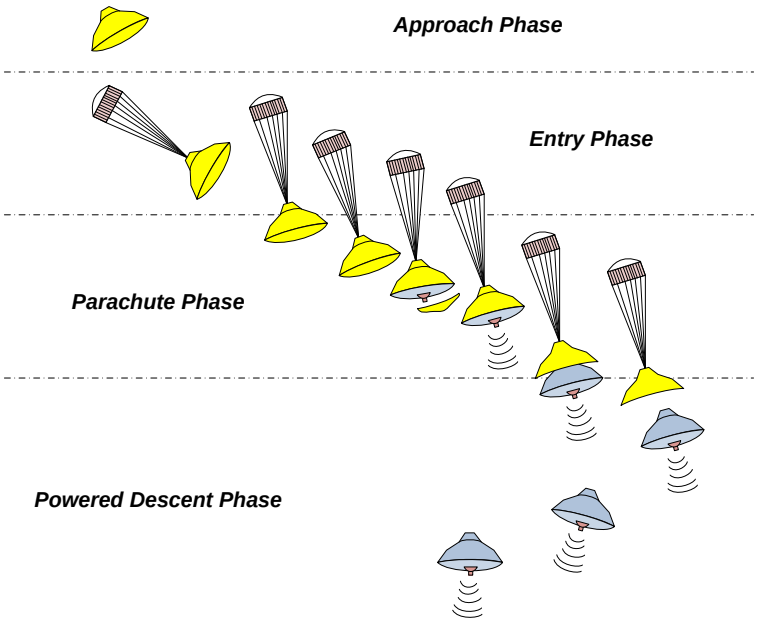

Figure 1. **Entry and descent scheme**

2.2 Powered Descent Phase

During the terminal phase starting at parachute release and lasting until thrusters are switched off, is usually obtained by appropriate orientation of the thrusters, which are rigidly connected to the vehicle. The main-thruster assembly is arranged to actuate a three degrees-of-freedom command (axial thrust, pitch and yaw torque) plus spin damping around the vehicle symmetry axis. Vehicle orientation (pitch and yaw) allows the axial thrust to be used for controlling the horizontal motion. Strategies for guidance and control to this phase were studied and implemented during the recent years (gravity-turn maneuver [30]–[34]). In Apollo-like guidance [13], [28], [35]–[39], the centre-of-mass (CoM)

trajectory is interpolated between initial and kinematic constraints through a 3D polynomial, thus becoming suitable to pinpoint landing [13]. The guidance of the MSL (Mars Science Laboratory [27], [37]), followed a modified Apollo guidance law using a fifth order polynomial law.

2.2.1 Guidance and control for the propulsion phase of planetary landing

In the propulsion phase of planetary landing, horizontal motion is obtained by tilting and aligning the axial thrust either to the opposite of the velocity vector or to the requested acceleration vector. The strategy of [7], [12] is assumed here, as it allows free horizontal motion and is preliminary to achieve accurate landing. Instead of designing a hierarchical guidance and control in which horizontal acceleration becomes the attitude reference, a unique control system is designed based on a fourth-order state equation per degree-of-freedom from the angular acceleration to the position coordinate.

To complete the tasks, axial thrust may be oriented either opposite to the current speed vector as in [30]–[34], or along the desired acceleration as in [13], [28], [35]–[39]. The previous approach is suitable to soft landing because it allows restricted horizontal diversions such as for escaping from back-shell and parachute trajectory as in [34]. In the latter approach the centre-of-mass (CoM) trajectory is interpolated between initial and terminal kinematic constraints through a 3D polynomial, thus becoming suitable to pinpoint landing. The guidance of the Mars Science Laboratory [27], which successfully landed on Mars in August 2012, employs a fifth order polynomial law for satisfying kinematic constraints at the powered phase.

Most of the studies focused on guidance problems, employs adaptive guidance to contrast disturbance [40], altitude measurement errors and target site modification. From this position simple feedback laws around the guidance trajectory are considered as sufficient, and are complemented with an attitude control around the reference trajectory imposed by CoM guidance. The solution applied here combines CoM and tilt dynamics, as the command acceleration of the horizontal motion [11], [12]. The suggested approach exploits

input-state linearized dynamics as in [41], [42], from the angular acceleration to horizontal position, and develops guidance, and control algorithms on the same state equations as suggested by the Embedded Model Control in [2], [3], [5], [43]. The modelling process has been outlined in [12].

2.3 Frames of References

In the planetary landing has become mandatory the uses of a set of references systems, these references frames are well studied in the literature [11], [12], [44], [45] in this section three of them are reviewed and the notation is unified.

2.3.1 Inertial frame of reference

The inertial reference frame $R=\left(C_p,\vec{i}_p,\vec{j}_p,\vec{k}_p\right)$ is centered on the planet center of mass CoM C_p.

2.3.2 Local Vertical Local horizontal frame of reference

The co-rotating local vertical local horizontal frame $R_f=\left(O,\vec{i}_l,\vec{j}_l,\vec{k}_l\right)$ is centered on the fixed surface point O and the axial direction $\vec{k}_l$, requirements are referred to local vertical local horizontal frame of reference. For these references the following assumptions are made:

- The vertical axis z_l is defined to be opposite to the planet gravity

$$\vec{k}_l=-\frac{\vec{g}}{g} \tag{2.1}$$

- The axis x_l is located on the plane orthogonal to the axis z_l in the same direction to the planet north. The axis y_l is defined as

$$\vec{j}_l=\vec{k}_l\times\vec{i}_l \tag{2.2}$$

- The origin O is rigidly connected to the landing target. The vertical axis z_l is defined to be opposite to the gravity in the same direction of the Zenith.

A transformation between the inertial frame of reference and the local frame of reference is introduced, and this is defined by two rotations with the latitude ϕ_L and longitude λ_L angles. The Figure 2 shows the inertial and local vertical local horizontal frame of references.

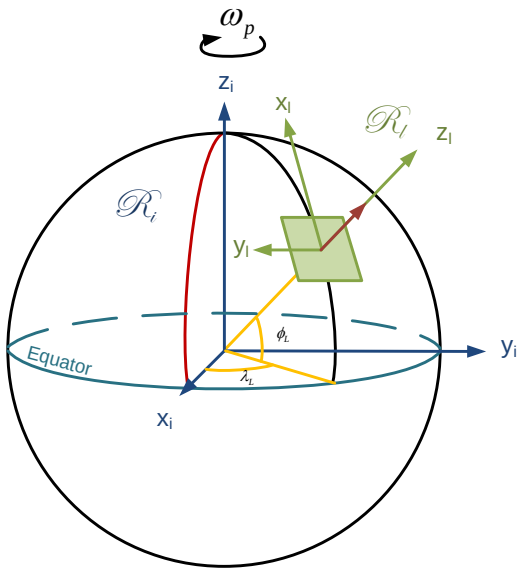

Figure 2. **Inertial and local frames of reference**

The transformation between the references systems is shown in the equation below

$$R_i^l = \begin{bmatrix} 0 & 0 & 1 \\ 0 & -1 & 0 \\ 1 & 0 & 0 \end{bmatrix} Y(\phi_L) Z(-\lambda_L) \tag{2.3}$$

2.3.3 Body frame of reference

The body frame of reference $R_b = \left(C, \vec{i}_b, \vec{j}_b, \vec{k}_b\right)$ is centered on the body CoM C and the axial direction $\vec{k}_b$ is directed opposite to the velocity vector $\vec{v}$

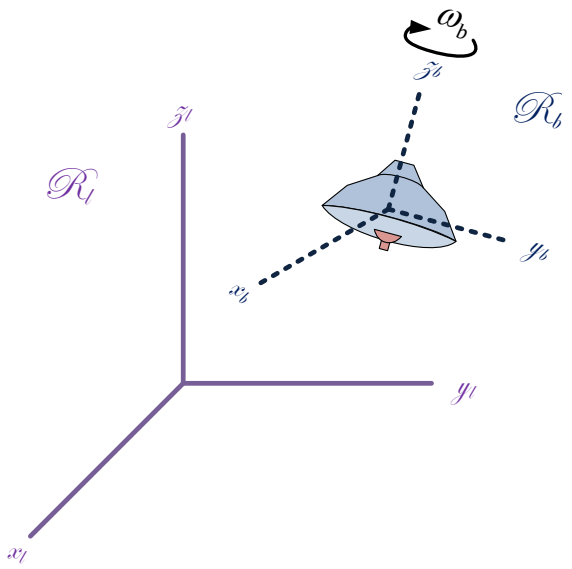

Figure 3. **Local and Body frame of references**

2.3.4 The descent equations

Guidance is a process that computes CoM and attitude courses from known initial conditions to target position and attitude, based on vehicle and environment models. Guidance trajectories become the references to be followed by feedback control law. In the propulsion descent the vehicle tilt determinates the horizontal force and therefore the acceleration. The guidance algorithms in [13], [17], [28], [35]–[39] split reference computation in a hierarchical method. They compute the desired CoM kinematic variables, position, velocity and acceleration, and design attitude control to track the reference computed by inverting the reference acceleration. The main advantage is a simplified control design, for this nonlinearity enters the transformation from CoM acceleration to attitude components.

The solution proposed in [11], [12] abandons the hierarchical approach by including attitude dynamics in the generation of the desired trajectory. To overcome the design problem posed by the nonlinear and variable link between attitude and CoM dynamics, input-state linearization as in [42] has been proposed and demonstrated in [12]. Linearization takes advantage of a bounded vehicle tilt such to accommodate localization

sensors (less than 1 radian) and of an appropriate Euler angle sequence, specifically 3-2-1, denoted by $\{\psi,\theta,\varphi\}$ The fourth-order multivariate (two degrees-of-freedom) differential equation of the tilt and horizontal motion, derived in [12], is the following

$$\begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\mathbf{v}}_x \\ \dot{\mathbf{q}} \\ \dot{\boldsymbol{\omega}}_x \end{bmatrix}(t) = \begin{bmatrix} 0 & I & 0 & 0 \\ 0 & 0 & a_{bz}I & 0 \\ 0 & 0 & 0 & I \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{v}_x \\ \mathbf{q} \\ \boldsymbol{\omega}_x \end{bmatrix} + \begin{bmatrix} 0 \\ b_m I \\ 0 \\ I \end{bmatrix} \mathbf{u}_x + \begin{bmatrix} 0 \\ \mathbf{d}_x \\ 0 \\ \mathbf{d}_m \end{bmatrix} \tag{2.4}$$

In (2.4) $\mathbf{x}$ and $\mathbf{v}_x$ denote the horizontal position and rate coordinates in the local vertical local horizontal frame $\left\{O,\vec{i}_l,\vec{j}_l,\vec{k}_l\right\}$. The local vertical local horizontal frame of reference is assumed as inertial because of the small planet rotation rate ω_p coupled with low altitude and speed during the descent phase.

Attitude and rate vectors are bounded nonlinear expressions of the Euler angles and of the body angular rate $\boldsymbol{\omega}_b$ as follows

$$\begin{aligned} \mathbf{q} &= \begin{bmatrix} q_x \\ q_y \end{bmatrix} = \begin{bmatrix} \cos\theta\sin\varphi \\ \sin\theta \end{bmatrix} \\ \boldsymbol{\omega}_x &= \begin{bmatrix} \cos\varphi & -\sin\theta\sin\varphi \\ 0 & \cos\theta \end{bmatrix} \begin{bmatrix} \cos\psi & -\sin\psi \\ \sin\psi & \cos\psi \end{bmatrix} \boldsymbol{\omega}_b \end{aligned} \tag{2.5}$$

The command angular acceleration $\mathbf{u}_x$ is a combination of command torques and gyro torques as a result of the linearization. The disturbance $\mathbf{d}_m$ and $\mathbf{d}_x$ include external perturbations and parametric uncertainty. The time varying gain is the axial acceleration of the vehicle, entering the vertical dynamics in (2.6). Equation (2.4) must be completed with vertical and spin rate dynamics as follows

$$\begin{aligned}\dot{z}(t) &= v_z(t)\\ \dot{v}_z(t) &= u_z(t) = \cos\varphi\cos\theta a_{bz}(t) - g + d_z(t)\\ \dot{\omega}_z(t) &= \alpha_z(t)\end{aligned} \tag{2.6}$$

g is the gravity acceleration and d_z encompasses disturbance and uncertainty. ω_z is the spin rate and α_z is the spin angular acceleration. Control and guidance algorithms are constructed around a discrete time (DT) version. The state variables are updated from navigation data.

3. OPTIMAL THRUSTERS DISPATCHING

3.1 Description

The configuration of the main thrusters has been studied in previous researches [7], [46] , a symmetric pyramidal thrusters geometry is analyzed with a generic case of n thrusters. They are equally spaced at a radius r_t from the z axis, at a vertical coordinate $-h_t$ with respect to the xy plane, with a cant angle β_t measured from the xy plane, and azimuth angles $\alpha_k, k = 1,...,m$, counted from the x-axis, see Figure 4.

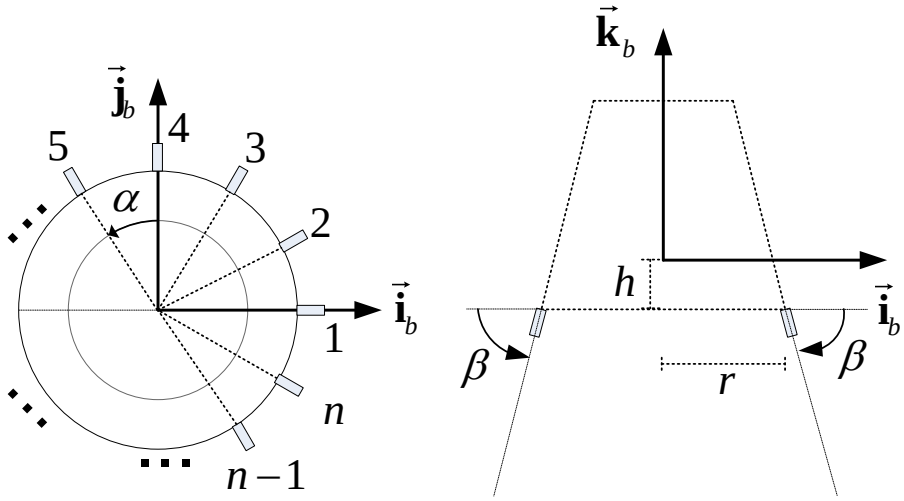

Figure 4. **Thruster geometry**

Consider a configuration with n thrusters, the assembly is organized into n $360/n$ apart clusters, they are ordered j=1,2,..,n. The magnitude force of each thrusters is denoted as $u_i, i = 1,..,n$, and total force thrusters vector is denoted as $\mathbf{f}$ The orientation matrix that includes the direction is defined as V. The equation (3.1) depicts the connection between the magnitudes u_i and total force.

$$\mathbf{f} = \begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix}_{3x1} = V \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix}_{3xn} \tag{3.1}$$

Hence the matrix V is denoted as

$$V = \begin{bmatrix} -s_\beta & -s_\beta c_\alpha & -s_\beta c_{2\alpha} & \cdots & -s_\beta c_{(n-1)\alpha} \\ -s_\beta s_0 & -s_\beta s_\alpha & -s_\beta s_{2\alpha} & \cdots & -s_\beta s_{(n-1)\alpha} \\ c_\beta & c_\beta & c_\beta & \cdots & c_\beta \end{bmatrix}_{nx3} \tag{3.2}$$
$$s_\beta = Sin(\beta), c_\beta = \cos(\beta), c_\alpha = \cos(\alpha), \alpha = 2\pi / n$$

The total force applied in the body frame of reference is assumed applied in the CoM, in order to obtain the torques the applied force point vectors are summarized in matrix A_v. Each column of the matrix A_v represents the position of each thruster in the body reference frame.

$$A_v = \begin{bmatrix} r & rc_\alpha & rc_{2\alpha} & \cdots & rc_{(n-1)\alpha} \\ rs_0 & rs_\alpha & rs_{2\alpha} & \cdots & rs_{(n-1)\alpha} \\ -h & -h & -h & \cdots & -h \end{bmatrix}_{3xn} \tag{3.3}$$

Total torque **m** can be obtained accumulating each singles torques, the torque dispatching matrix is made by the moments of the directions.

$$\mathbf{m} = \begin{bmatrix} m_x \\ m_y \\ m_z \end{bmatrix}_{3x1} = M_t \mathbf{u}, \mathbf{u} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix}_{nx1} \tag{3.4}$$

Where the matrix C_t is denoted as

$$M_t = d \begin{bmatrix} 0 & s_\alpha & s_{2\alpha} & \cdots & -s_{(n-1)\alpha} \\ -1 & -c_\alpha & -c_{2\alpha} & \cdots & -c_{(n-1)\alpha} \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix}_{3xn}, d = (rc_\beta - hs_\beta) \tag{3.5}$$

Note the first two lines of matrix M_t and V are linearly dependent, hence, it is not possible to control the total force and moment at the same time with this configuration.

3.2 Inverse Law

Equations (3.4) and (3.1) show that thrusters are arranged to drive a three degrees-of-freedom command (axial thrust, pitch and yaw torque), the number of main thrusters allows to define an optimization problem, where the functional can be exploit in order to reduce

the propellant consumption, or avoid engines constrains. In this section in order to guarantee fast computational process a quadratic functional is selected. A pseudo inverse law is implemented, additionally a version of the recursive minimum quadratic algorithm is implemented in order to avoid the thrusters constrains taking advantage of the worst-case analysis. At the end some partial results are shown.

3.2.1 Unconstrainted Thrusters Analysis

The command of a spacecraft as it is mentioned before has only three degrees-of-freedom, a vector $\mathbf{f}_{zm}$ command is defined in the equation (3.6), where the matrix B_{zm} represents the relationship between x,y torques, z force and the single thrusters magnitudes,

$$\mathbf{f}_{zm} = \begin{bmatrix} f_z \\ m_x \\ m_y \end{bmatrix}_{3x1} = B_{zm}\mathbf{u}, \mathbf{u} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix}_{nx1} \tag{3.6}$$

The thrusters vector $\mathbf{u}$ is a decision variable, the objective is to minimize the norm 2 of the vector, therefore least squares method is used. The next steps are executed in order to obtain a fast algorithm to define the magnitude force value of the thrusters, also with straight implementation.

The matrix B_{zm} can be obtained from matrices V and M_t in (3.8), from this a new sized known matrix G is defined,

$$\begin{aligned} &G = B_{zm}B_{zm}^T \\ &rank(B_{zm}B_{zm}{}^T) = rank(B_{zm}{}^T B_{zm}) = rank(B_{zm}) = 3 \end{aligned} \tag{3.7}$$

and

$$B_{zm} = d\begin{bmatrix} c_\beta / d & c_\beta / d & c_\beta / d & \cdots & c_\beta / d \\ 0 & s_\alpha & s_{2\alpha} & \cdots & -s_{(n-1)\alpha} \\ -1 & -c_\alpha & -c_{2\alpha} & \cdots & -c_{(n-1)\alpha} \end{bmatrix}_{3xn} \tag{3.8}$$

The optimal problem is solved through the pseudo inverse strategy [47] the solution without constraints is shown in the equations below

$$\mathbf{u} = B_{zm}^{T} G^{-1} \mathbf{f}_{zm} \tag{3.9}$$

where G and G^{-1} follows the expressions

$$G = d^{2} \begin{bmatrix} nc_{\beta}{}^{2} / d^{2} & 0 & 0 \\ 0 & n/2 & 0 \\ 0 & 0 & n/2 \end{bmatrix}_{3x3} \tag{3.10}$$

$$G^{-1} = \frac{1}{d^{2}} \begin{bmatrix} d^{2} / \left(nc_{\beta}{}^{2} \right) & 0 & 0 \\ 0 & 2/n & 0 \\ 0 & 0 & 2/n \end{bmatrix}_{3x3} \tag{3.11}$$

A very interesting result is obtained when the expression (3.9) is itemized

$$\mathbf{u} = \frac{1}{d} \begin{bmatrix} d/\left(nc_{\beta} \right) & 0 & -2/n \\ d/\left(nc_{\beta} \right) & 2s_{\alpha}/n & -2c_{\alpha}/n \\ d/\left(nc_{\beta} \right) & 2s_{2\alpha}/n & -2c_{2\alpha}/n \\ \vdots & \vdots & \vdots \\ d/\left(nc_{\beta} \right) & 2s_{(n-1)\alpha}/n & -2c_{(n-1)\alpha}/n \end{bmatrix}_{nx3} \mathbf{f}_{zm} \tag{3.12}$$

Combining (3.2) and (3.12), the relation between the commands and forces in (3.13) was obtained which shows connections that can be considered as a known disturbances [5] and therefore they are part of the spacecraft dynamics described in (2.6).

$$\begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix} = \begin{bmatrix} 0 & 0 & \dfrac{sb}{d} \\ 0 & -\dfrac{sb}{d} & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} f_z \\ m_x \\ m_y \end{bmatrix} \tag{3.13}$$

3.2.2 Constrainted Thrusters Analysis

3.2.2.1 Problem statement

In order to determinate which thrusters is closer to saturation, a general version of (3.12) is used,

$$\begin{aligned} &0 < i < n-1 \\ &u_k = \frac{1}{d}\begin{bmatrix} \frac{d}{nc_\beta} & \frac{2s_{\alpha k}}{n} & \frac{-2c_{\alpha k}}{n} \end{bmatrix}_{3xn} \mathbf{f}_{zm} \end{aligned} \tag{3.14}$$

Each u element is a dot product between each $B_{zm}^T G^{-1}$ row and the vector $\mathbf{f}_{zm}$

$$\frac{f_z}{nc_\beta} + \frac{2s_{\alpha i} m_x}{dn} - \frac{2c_{\alpha i} m_y}{n} = u_k \tag{3.15}$$

The command u_k is divided in two terms $u_k = u_f + u'_k$, the first one depends on the axial force $u_f = f_z / nc_\beta$, this remains constant for all thrusters, the other equation part is defined by the torques m_x and m_y, on equation (3.16) the torques are changed in a magnitude and phase representation.

$$\begin{aligned} &\mathbf{m}_{xy} = \begin{bmatrix} m_x \\ m_y \end{bmatrix} = \left\| \mathbf{m}_{xy} \right\| \begin{bmatrix} \cos(\phi) \\ sen(\phi) \end{bmatrix} \\ &u'_k = \mathbf{m}_{xy}{}^T \frac{2}{nd} \begin{bmatrix} sin(\alpha(k-1)) \\ -\cos(\alpha(k-1)) \end{bmatrix}, 1 \le k \le n, \alpha = 2\pi / n \end{aligned} \tag{3.16}$$

For each k-th thruster, the command is defined between the angle of the moment required and the thruster location, a geometrical expression is shown below

$$\begin{aligned} u'_k &= \frac{2\left\| \mathbf{m}_{xy} \right\|}{nd} \left(\sin\left(2\pi(k-1)/n\right)\cos(\phi) - \cos\left(2\pi(k-1)/n\right)\sin(\phi) \right) \\ u'_k &= \frac{2\left\| \mathbf{m}_{xy} \right\|}{nd} \sin\left(2\pi(k-1)/n - \phi\right) \end{aligned} \tag{3.17}$$

The worst case depends on the k-th thruster selected, this case is found when (3.17) is maximised. Equation (3.18) shows the result.

$$\begin{aligned} &2\pi(k-1)/n - \phi = \pi/2 \\ &k = n/4 + n\phi/(2\pi) + 1 \end{aligned} \tag{3.18}$$

The number k has discrete values but the angle ϕ takes continuous values, therefore the maximum force is applied in the closest thrusters to the result (3.18). The thrusters constraints are defined by

$$\begin{aligned} u_{\max} &\geq f_z/(nc_\beta) + 2\left\|\mathbf{m}_{xy}\right\|/(nd) \\ u_{\min} &\geq f_z/(nc_\beta) - 2\left\|\mathbf{m}_{xy}\right\|/(nd) \end{aligned} \tag{3.19}$$

While a thrusters is not saturated the result (3.9) is applicable. This is posed as a quadratic programming problem with linear constrains. The objective function is,

$$Minimize \ \left\|\mathbf{u}\right\| \tag{3.20}$$

Constrains from (3.6) and (3.19) are summarized next

$$\begin{aligned} B\mathbf{u} &= \mathbf{f}_{zm} \\ \mathbf{u} &< \mathbf{u}_{\max} \\ \mathbf{u} &> \mathbf{u}_{\min} \end{aligned} \tag{3.21}$$

3.2.2.2 Thrusters Constrains Analysis with force Reduction

In order to avoid constraints a recursive least squares algorithm is performed, the vector is divided in two vectors. The first one $\mathbf{u}_1$ is composed by the thrusters that are free to take the decision. The second $\mathbf{u}_2$ is formed by the thrusters which using the solution (3.9) overcome the bounds in the equation (3.19), these are forced to saturate, therefore the values of $\mathbf{u}_2$ are known, hence the values of $\mathbf{u}_1$ can be found by solving the next problem.

$$\begin{aligned} \mathbf{u} &= \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{bmatrix} \\ B_{zm1}\mathbf{u}_1 + B_{zm2}\mathbf{u}_2 &= \mathbf{f}_{zm} \\ B_{zm1}\mathbf{u}_1 &= -B_{zm2}\mathbf{u}_2 + \mathbf{f}_{zm} \end{aligned} \tag{3.22}$$

The solution (3.9) is applied in order to solve equation (3.22), this Is possible until given the condition $rank(B_{zm}) = 3$, when the rank decrease, a critical choice is made, the stability

of the spacecraft is connected with the attitude, the tilt and therefore the moment in any case is able to guide the vehicle out to the saturation. The alternative explores here in order to avoid the saturation is to decrease the axial force f_z. When f_z is decreased, the system remains into a constrained hyper plane and then is possible to apply the result (3.12) with a rank 2.

3.2.3 Simulation results

A Matlab-Simulink model was implemented in order to test the algorithms developed above. A test for a spacecraft with 12 main thrusters with the following features was implemented; $\beta = \pi / 4$, radius $r = 2m$, and a thrusters height $h = 1m$, Figure 5,Figure 6 and Figure 7 shows the total axial force and the total moments requested , and the thrusts obtained for each engine.

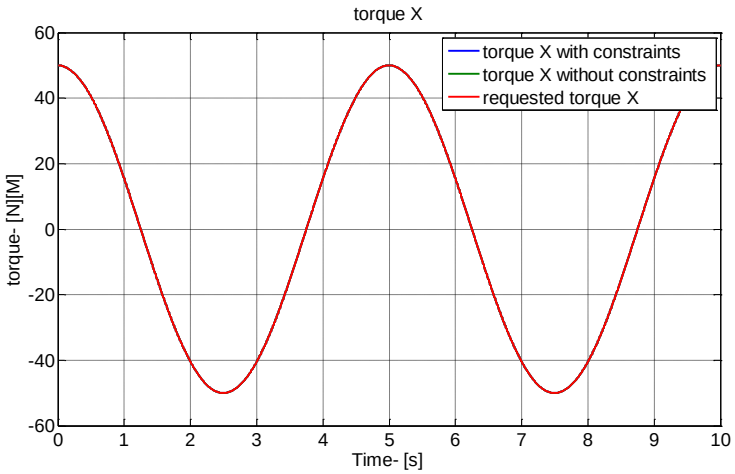

Figure 5. **X Moments**

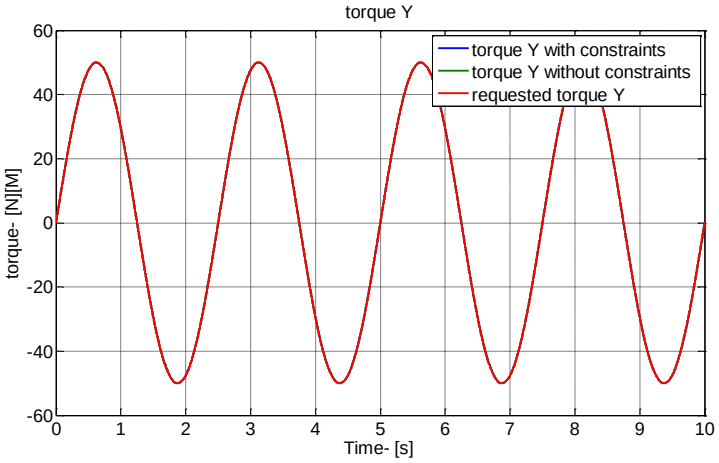

Figure 6. **Y Torque**

It is important to note that when the force and the applied torque cannot be guaranteed, the axial force is reduced. The green line in Figure 7 depicts the output force when bounds are surpassed.

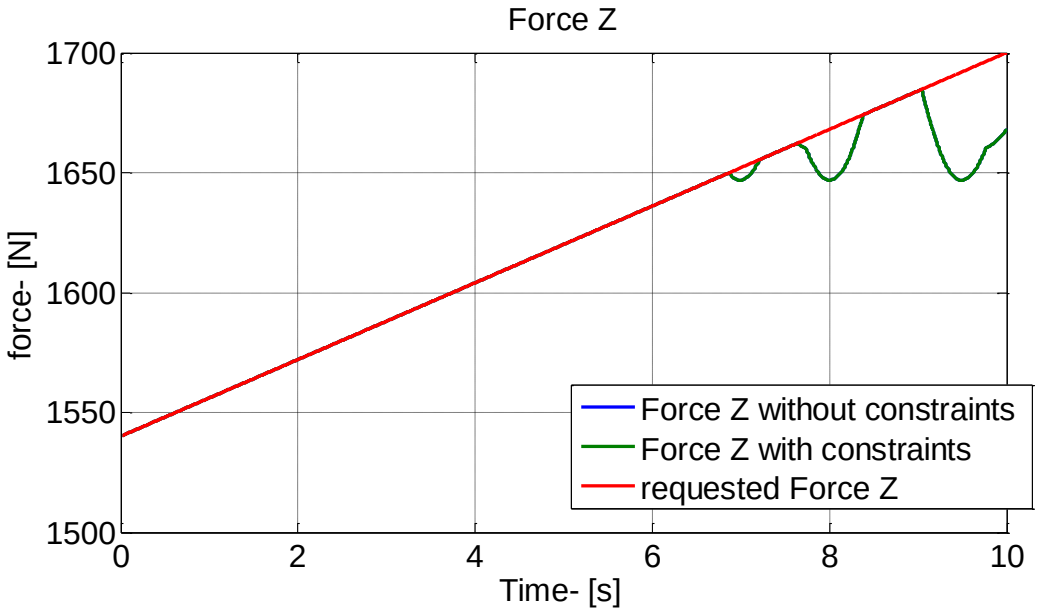

Figure 7. **Axial Force**

Figure 8 shows the force of each thruster, a saturation is obtained when the requested axial force overcome the constraint (3.19). The optimal dispatching algorithm is applied until only 2 thrusters are used to maintain the requested moment (Figure 8), in that case the axial force is reduced to guarantee the moment requested. Similar results are obtained on [48] for n-rotor dispatching.

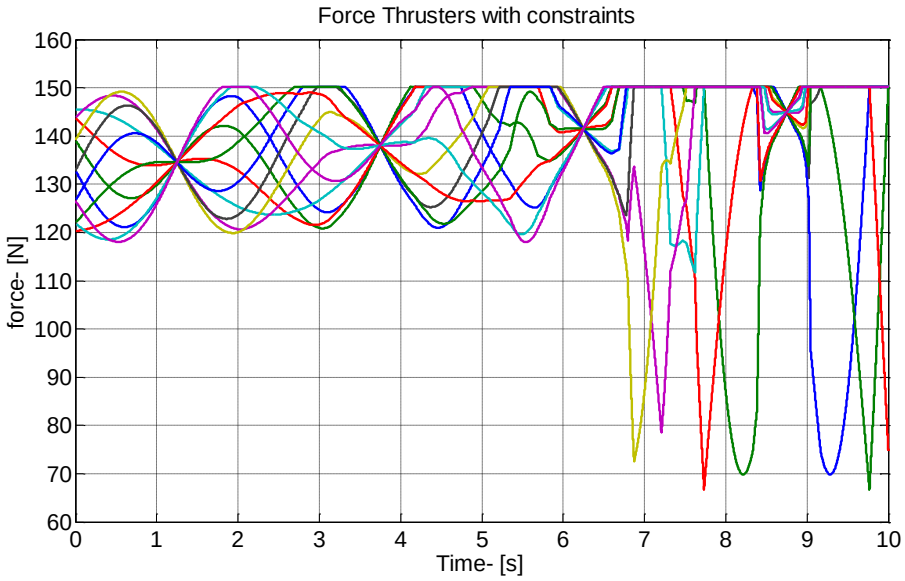

Figure 8. **Thrusters Force**

4. EMBEDDED MODEL CONTROL FOR PLANETARY TERMINAL DESCENT PHASE

The research shows the control design around the reference trajectory (tilt and position) given by the guidance that takes advantage of the quasi linearization based on embedded model control method [11], [12]; the first part of this dissertation is restricted to closed-loop control strategies.

4.1 Embedded Model Control EMC

Robust control design is dedicated to guarantee the closed-loop stability of a model-based control law in the presence of parametric uncertainties[49]. This law uses diverse methodologies that are derived from non linear models. Stability is guaranteed by introducing some coefficients and reducing the feedback control effort. Embedded Model Control (EMC) [1]–[3], [5], [6], [43] illustrates that a control law has to and can be kept without modifications in the case of uncertainty, if the controllable dynamics is complemented with a disturbance dynamics capable of real-time encoding the different uncertainties that affect the embedded model (EM). The disturbance state is updated in real-time by a noise input vector, which is estimated from the model error only. Model error $\mathbf{e}$ is the sole available measure of the uncertain discrepancies, i.e. it is the difference between plant and model output. Feedback control reduces output sensitivity to discrepancies. Sensitivity may be further abated by explicitly rejecting disturbance, Disturbance dynamics is widely studied in the literature [50], [51]. Model error can be elaborated and accumulated in a state vector $\mathbf{x}_d$ (disturbance state), ready to correct $\mathbf{x}_c$. Formally, an observable input-output dynamics $\mathbf{D}$ must be built, from an input noise $\mathbf{w}$ to an output dynamics $\mathbf{d}$, the latter forcing $\mathbf{M}$ in parallel to $\mathbf{u}$. As a result, $\mathbf{x}_d$ encodes the past accumulated discrepancies, whereas $\mathbf{w}$ encodes the past and future independent uncertainty capable of updating $\mathbf{x}_d$. Independent of future derives from causality, whereas independence of past answers the principle of not delaying disturbance updating. For such reasons $\mathbf{w}$, should be treated as a set of arbitrary and bounded zero-mean signals, flat spectrum in the frequency

domain, and statistically as a bounded-variance discrete-time white noise. In other terms, no state equation exists relating past to future of $\mathbf{w}$.

Two alternative mechanisms can generate noise: i) pseudo-random extraction, ii) estimation from a correlated realization. The former would respect noise statistical properties, the latter, to be adopted, reveals the residual discrepancies that are hidden in the model error to the benefit of the embedded model, as it can be driven to approach the plant and to bound $|\mathbf{e}|$. Complexity and uncertainty of discrepancies may suggest abandoning the statistical framework in favour of a bounded arbitrariness, which entails command independence.

Appropriate separation of the uncertainty components into low and high frequency domains by the noise estimator allows stability recovery and guarantees the rejection of the low frequency uncertainty components. For this research an emphasis is given to the control unit. The embedded model in section 4.2 is forced by two input vectors: $\mathbf{u}(i)$ is known since it is computed at any step i by the control unit, $\mathbf{w}(i)$ is defined to be unknown and unpredictable. The uncertainty design include the noise estimator, as the model error may convey uncertainty components (parameters, cross-couplings, neglected dynamics) which are command-dependent and thus are prone to destabilize the controlled plant, into the embedded model (Figure 9).

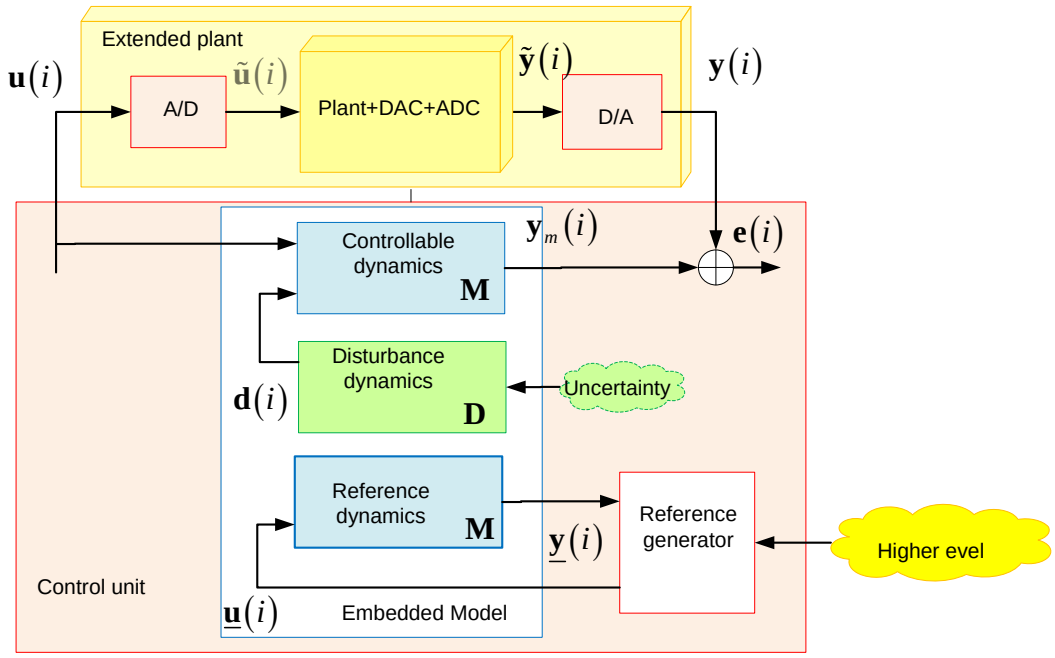

Figure 9. **The plant and the parallel embedded model as the core of the control unit.**

4.2 Horizontal Embedded model

A model can run in parallel and synchronous (real-time) with the plant under the same admissible command $\mathbf{u}$ as in Figure 9. The principle is seminal to subsequent formulation, as well as to control architecture, as it suggests that control units shall develop around the real-time model, henceforth indicated as the 'embedded model'. Restricting to computer-based control, a real-time model can only be discrete time and state variable[52], implying a time unit T and a state $\mathbf{x}_c$ must be defined.

The embedded model is the ensemble of the discrete time version of (2.4), referred to as controllable dynamics, and disturbance dynamics, in charge of expressing the unknown time evolution of the disturbance $\mathbf{d}_m$. The controllable dynamics links the command vector $\mathbf{u}_x$ to the model output $\mathbf{y}_m$. The disturbance dynamics links the disturbance vectors to an arbitrary signal vector $\mathbf{w}$ referred to as noise. The strategy allows pinpoint landing. As such, tilt angles (pitch and yaw) become proportional to the horizontal acceleration. Instead of designing a hierarchical guidance and control in which horizontal acceleration becomes

the attitude control target, a unique control system can be designed based on the fourth order dynamics from angular acceleration to position.

4.2.1 Controllable dynamics

To compute a smooth angular acceleration $\mathbf{u}_x$, the latter vector is treated as a state variable, which asks for a new command vector $\mathbf{s}_x$, called jerk. As a further constraint, all the state variables are given the same measurement units, which are guaranteed by scaling them times the time unit T of the control system, which is fixed by the thruster actuation time unit. Since a property of (2.4) is that the horizontal components are completely decoupled, a single scalar component of $\mathbf{x}$ simplified to x will be treated hereafter. The controllable state vector is defined as

$$\begin{aligned} &\mathbf{x}_c = \begin{bmatrix} x & \Delta v_x & q & \Delta\omega_x & \Delta\alpha_x \end{bmatrix}^T \\ &\Delta v = v_x T,\ \Delta\omega_x = \omega_x T,\ \Delta\alpha_x = u_x T^2 \end{aligned} \tag{4.1}$$

The discrete time state equation is obtained by integrating (2.4) along the time unit T and holds

$$\mathbf{x}_c(i+1) = A(i)\mathbf{x}_c(i) + Bs(i) + \mathbf{d}(i) \tag{4.2}$$

Matrices in (4.2) are the following:

$$\begin{aligned} A &= \begin{bmatrix} 1 & 1 & \alpha/2 & \alpha/6 & \alpha/24+\beta/2 \\ 0 & 1 & \alpha & \alpha/2 & \alpha/6+\beta \\ 0 & 0 & 1 & 1 & 1/2 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix},\ B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \\ \alpha &= a_{bz}(iT)T^2, \beta = b_m \end{aligned} \tag{4.3}$$

The measurements of the embedded model state variables are ‘pseudo measurements’ provided by attitude, angular rate, position and velocity that are estimated by the navigation algorithm. Only the angular acceleration u_x is not measured. The output equation is therefore

$$\mathbf{y}_m = \begin{bmatrix} I_4 & 0 \\ 0 & 0 \end{bmatrix} \mathbf{x}_c + \mathbf{e}(i) \tag{4.4}$$

4.2.2 Disturbance dynamics

Disturbance dynamics expresses the disturbance vector **d** in (4.2) as a combination of a third order state vector $\mathbf{x}_d$, of a fifth order noise vector **w** and of the vector $\mathbf{d}_k$ of the known interconnections coming out from the input-state linearization. The noise vector **w** is the only input driving $\mathbf{x}_d$. A linear, time invariant combination is sufficient

$$\mathbf{d}(i) = H_c \mathbf{x}_d(i) + G_c \mathbf{w}(i) + \mathbf{d}_k(i) \tag{4.5}$$

with matrices

$$H_d = \begin{bmatrix} \frac{1}{2} & \frac{\alpha}{24} + \frac{\beta}{2} & 0 \\ 1 & \frac{\alpha}{6} + \beta & 0 \\ 0 & 1/2 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \; G_c = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{4.6}$$

Because of the structure of G_c in (4.6), **w** does not directly affect all the state variables in(4.1), but only $v_x(i+1)$ and $\omega_x(i+1)$. A scheme of this kind looks coherent with the absence of noise in the chain from acceleration to position (noise design as [53]). The third order state equation is

$$\begin{aligned} \mathbf{x}_d(i+1) &= A_d \mathbf{x}_d(i) + G_d \mathbf{w}(i) \\ \mathbf{x}_d &= \begin{bmatrix} d_x & d_q & s_q \end{bmatrix} \end{aligned} \tag{4.7}$$

The first component, expressing a random drift, refers to $\mathbf{d}_x$ in(2.4), whereas the second and third components -second-order random drift - refer to $\mathbf{d}_m$ in(2.4). Matrices in (4.7) hold

$$A_d = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix},\ G_d = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.8}$$

4.3 The guidance algorithm

Guidance is computed under the restrictive assumption of a uniform vertical deceleration. Extension to hovering and ascent as in [14], [17], [54] for increasing horizontal motion is an extension under development, which may be essential for hazards avoidance. Vertical guidance drives the horizontal guidance through the reference $\underline{\alpha}(i)$. Assuming uniform deceleration, initial altitude $h(0)$, velocity $v_z(0)$ and descent duration t_f are related. For instance given $v_z(0)$ and t_f the initial altitude to start from is obtained. Vertical guidance then provides the reference altitude $\underline{z}$, velocity $\underline{v}_z$ and reference acceleration $\underline{u}_z$. The horizontal guidance algorithm minimizes a norm of the jerk s, thus limiting acceleration slew rate. Given $\underline{u}_z$ the reference gain $\underline{\alpha}$ can be computed as

$$\underline{\alpha}(i) = \frac{\underline{u}_z(i)T^2}{\sqrt{1-\left(\underline{q}_x^2(i)+\underline{q}_y^2(i)\right)}} \tag{4.9}$$

thus depending on the horizontal guidance. In addition to jerk, the horizontal guidance minimizes the energy of the tilt angles in (4.9), which is related to propellant consumption. Optimization is constrained by tilt bound, and is iterated to accommodate the nonlinearity in (4.9). Guidance can be adapted to a target site update until a minimum altitude is reached.

4.4 Control law

Following EMC, the control law of each horizontal has the following form

$$s(i) = \underline{s}(i) + K\left(\underline{\mathbf{x}} - \mathbf{x}_c - Q\mathbf{x}_d\right) - M\mathbf{x}_d \tag{4.10}$$

which is the sum of the reference jerk $\underline{s}$, of a feedback control proportional to tracking error $\underline{\mathbf{x}} - \mathbf{x}_c - Q\mathbf{x}_d$, and of the disturbance state $\mathbf{x}_d$ to be rejected. The tracking error includes the disturbance state, as the latter affects an intermediate state (the horizontal

acceleration) of the fifth order chain from jerk to horizontal position. Matrices M and Q in (4.10) are juts imposed by the embedded model in (4.2) and (4.7) using the Sylvester-type matrix equation

$$\begin{bmatrix} H_c + QA_d \\ 0 \end{bmatrix} = \begin{bmatrix} A & B \\ F & 0 \end{bmatrix} \begin{bmatrix} Q \\ M \end{bmatrix} \qquad (4.11)$$
$$F = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

It is straightforward to find the following solutions

$$Q = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1/\alpha & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \end{bmatrix} \qquad (4.12)$$
$$M = \begin{bmatrix} 0 & 0 & 1 & 0 & 1 \end{bmatrix}$$

The feedback matrix $K = \begin{bmatrix} k_x & k_v & k_q & k_\omega & k_u \end{bmatrix}$ is related with the desired eigenvalues of the closed-loop control, which in turn define the coefficients of characteristic polynomial

$$P(\gamma) = \gamma^5 + c_4\gamma^4 + c_3\gamma^3 + c_2\gamma^2 + c_1\gamma + c_0 \qquad (4.13)$$

The five gains are uniquely obtained by solving the following equalities given α and β and the coefficients of (4.13).

$$\begin{aligned} &c_4 = k_u,\ c_3 = k_\omega + \frac{k_q}{2} + \left(\frac{\alpha}{6} + b\right)k_v + \left(\frac{\alpha}{24} + \frac{\beta}{2}\right)k_x \\ &c_2 = k_q + \alpha k_v + \left(\frac{7\alpha}{12} + \beta\right)k_x,\ c_1 = \alpha k_v + \frac{3}{2}\alpha k_x \\ &c_0 = \alpha k_x \end{aligned} \qquad (4.14)$$

To compute (4.10) the current one-step predictions of the controllable state $\mathbf{x}_c$, and the disturbance state $\mathbf{x}_d$ are required. They are obtained from the predicted 3D position $\mathbf{r}$, velocity $\mathbf{v}$, attitude $\mathbf{q}$ (expressed as a quaternion) and angular rate $\boldsymbol{\omega}$ of the body, output of the navigation algorithm. A nonlinear transformation $P(\cdot)$ converts the navigation pseudo measurements into measurements $\mathbf{y}$ compatible with the model output $\mathbf{y}$ in (4.4). The

model error $\mathbf{e} = \mathbf{y} - \mathbf{y}_m$ is employed as in Kalman filters to estimate the noise vector $\mathbf{w}$ in (4.5) and to update the disturbance and controllable states $\mathbf{x}_c$ and $\mathbf{x}_d$ (output of the embedded model block). The embedded model state variables and the reference state and jerk $\underline{\mathbf{x}}$ and $\underline{\mathbf{s}}$ enter the control law block implementing (4.10). The output is the commanded jerk $\mathbf{s}$ which is integrated to provide the angular acceleration $\mathbf{u}_x$. The conversion from $\mathbf{u}_x$ to the thrust vector and vice versa passes through a nonlinear transformation $S(\cdot)$ and the thruster dispatching law. The thruster vector is then converted back to $\mathbf{u}_x$, the embedded model command.

5. HAZARD DETECTION AND AVOIDANCE

5.1 Piloting

In order to select the best landing site, the position target in the local reference system can change. The piloting is a process computed in real time that sets or changes the landing target position in order to avoid obstacles and to define a safe landing region. The piloting is the input on the guidance and in general this topic is treated separately from the GNC. In this section a piloting based on hazard detection and avoidance is presented, and it is connected with the guidance and control algorithms based on EMC theory [7], [11], [12]. Figure 10 shows how the piloting is included in the GNC scheme.

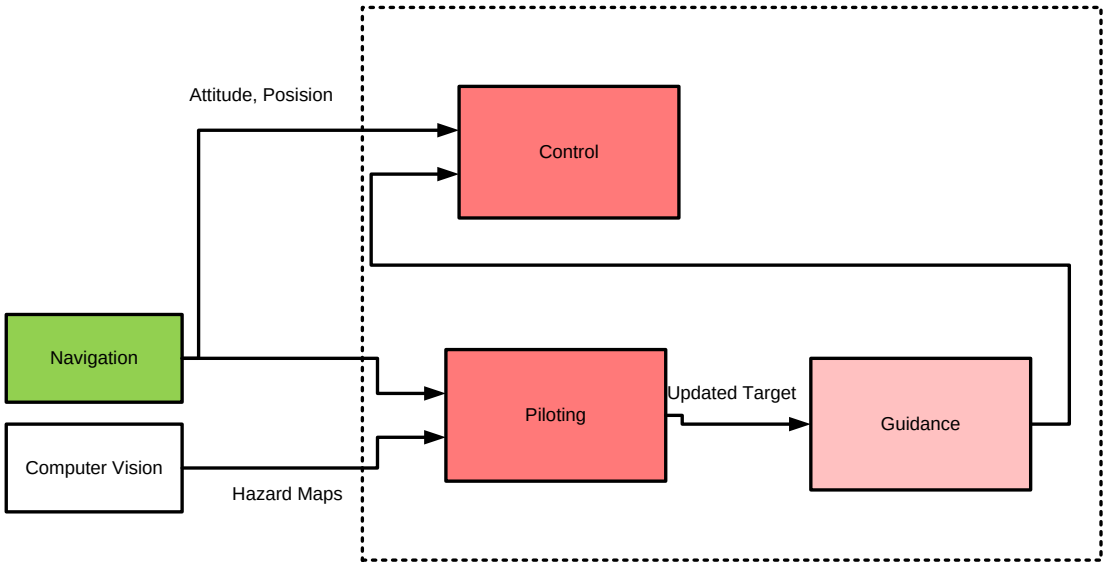

Figure 10. **Piloting**

5.2 Hazard Maps

The hazard maps (HMs) have been studied in recent years [18], [55]–[60]. HMs are divided in two components, the first one is a constant component due to topographic elements (slopes, rocks, inter alia) and other due to not permanent elements (shadows), all are provided by a vision based process [18], [19], [55], [57] that uses the camera on board with a field of view (FOV). FOV is determinate by the camera's angle of view δ and the vehicles attitude.

Figure 11 and Figure 12 shows examples of HMs used in this research [18]. The camera provides the image in the camera frame of reference, and navigation provides the position and attitude, with this information the image processing generates as output a projected HM in the local vertical local horizontal frame of reference. The projected HM are assumed as piloting inputs.

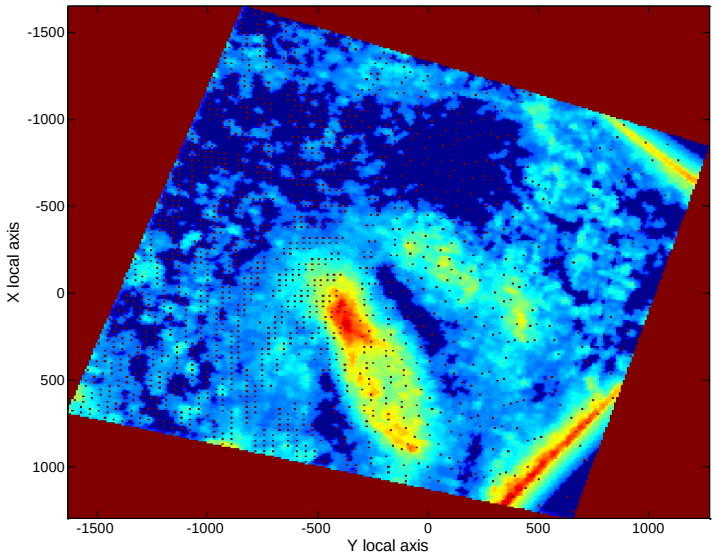

Figure 11. **Projected map at 2000 m altitude.**

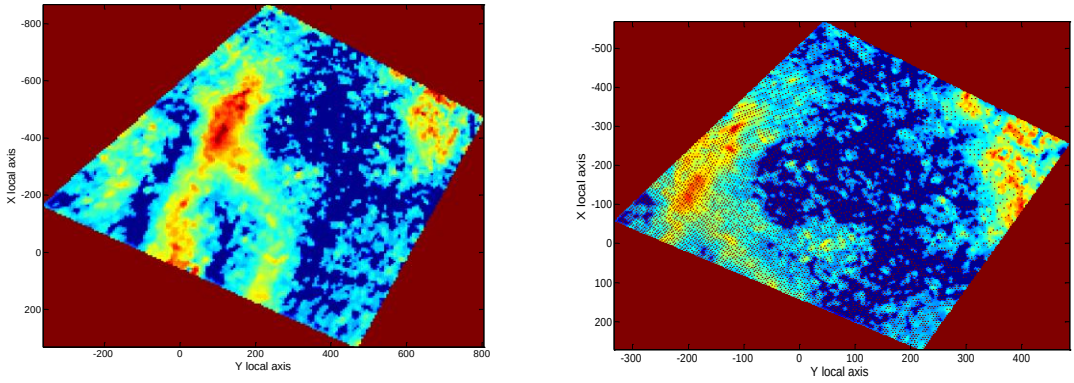

Figure 12. **Projected map at 700 m and 500 m altitude, with different local coordinates.**

The projected HM (Figure 11) is a matrix of elements that represents the risk that exists when a spacecraft lands on a particular area (0=safe, 1 = unsafe), these HMs are in the

vertical local horizontal local frame of reference, the size of the map is defined by the camera. For the analysis of the HM the following definitions are made;

- *Safety level* $s(i,j)$ (SL): it is the value between 0 and 1 that defines the landing quality of a pixel on the projected hazard map a safe landing site should satisfy that the surface slope must be below 15 degrees and probability of landing on a rock greater than 33 cm high should be less than about 1% [60],
- *Safety threshold* $s_{\max}$: only the pixels with a safety level $s(i,j) \le s_{\max}$ (acceptable level, AL) are candidates to be landing site.

It is assumed that the lander CoM C is located in the local vertical local horizontal frame before piloting and guidance functions start. When guidance and piloting start, the origin of the local vertical local horizontal frame may be outside the camera FOV. Guidance will direct the lander to the approach direction and to the landing point. If the landing point is outside of the propellant ellipse, piloting function will find an acceptable site inside propellant ellipse and close to the target landing point.

Target landing ellipse T is the predefined landing ellipse of the mission [8], [27]; it defines the target frame. It holds

$$\begin{aligned} &\begin{bmatrix} r_x - c_x & r_y - c_y \end{bmatrix}^T S^2 \begin{bmatrix} r_x - c_x \\ r_y - c_y \end{bmatrix} = 1 \\ &S^2 = \begin{bmatrix} s_x^2 & s_{xy} \\ s_{xy} & s_y^2 \end{bmatrix} > 0 \\ &c_x = c_y = 0 \end{aligned} \tag{5.1}$$

The ellipse centre is $\mathbf{c}_O = \{c_x = 0, c_y = 0, c_z = 0\}$. The semi-axes are $a > 0, b > 0$ [m] and the angle $|\psi| = 0$ of the 'main' axis with $\vec{t}_1$. The semi-major axis is the first target frame axis. The ellipse parameters are related to parameters in (5.1) by

$$\tan(2\psi) = \frac{2s_{xy}}{s_x^2 - s_y^2} = 0, s_x^2 \geq s_y^2$$
$$\frac{1}{a^2} = \frac{s_x^2 + s_y^2}{2} + \frac{s_{xy}}{\sin(2\psi)}; \frac{1}{a^2} = s_x^2, s_{xy} = 0 \quad (5.2)$$
$$\frac{1}{b^2} = \frac{s_x^2 + s_y^2}{2} - \frac{s_{xy}}{\sin(2\psi)}; \frac{1}{b^2} = s_y^2, s_{xy} = 0$$

The dimensions of the target site should be related to the landing uncertainty. To this end the minimum (size) landing site should be defined a priori, by fixing the lower limit $a_{\min}$ of a and b. A sub-task aiming to such a computation has been added, but it is not essential (last priority). For now $a_{\min} \leq 100\ \mathrm{m}(3\sigma)$. The target site ellipse may have $a >> b$ which suggests that the landing site is approached along the semi-major axis direction.

To find the landing site, the hazard map is further compressed into average squares. Currently the number of macro-pixels aggregated is $m \times m = 2 \times 2$. From the altitude of about 500 m, the edge (pixel width) of an AS is about 4 m. It is coherent with the size of the map in Figure 11, and it corresponds to the landing platform footprint. At an altitude of about 4000 m the edge length is about 33 m and it corresponds to the landing uncertainty (1 sigma), $w(4000) \cong a_{\min} / 3 = 33\ \mathrm{m}$. The average square map (ASM) is a map with square pixels whose width is a real length in the target frame. The pixel width $w(h)$ of the ASM varies with the altitude. The process is shown in Figure 13.

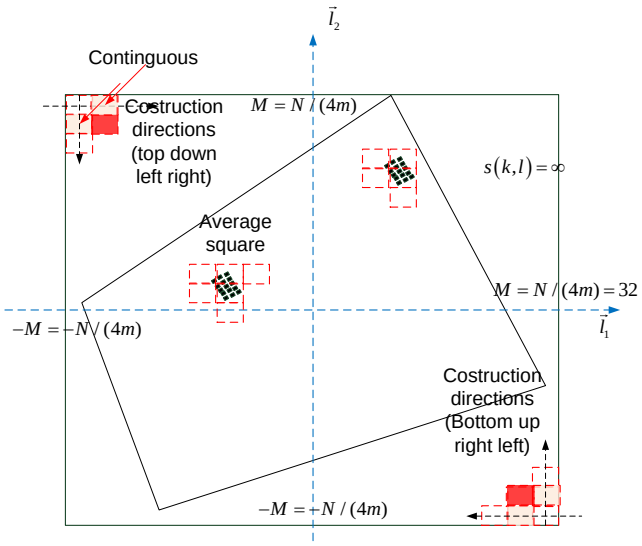

Figure 13. **Average square construction.**

Landing site L is the output of the piloting function, and it is defined as an ellipse whose average square (AS) has acceptable safety level (below the threshold). The equation of the ellipse is the same as (5.1). The landing site is selected to be close to the target site and inside the propellant ellipse (see Figure 14). Two landing sites are selected: (i) the coarse landing site is selected just after the guidance is activated and the lander camera is directed toward the target site; it occurs at about 4000 m and the resolution is larger than 30 m to avoid coarse hazards. (ii) the fine landing site is obtained when after the coasting (sailing) phase, the lander camera points again to the coarse landing site and a fine landing site is obtained with resolution about 5 m. During the acceleration phase and the early coast phase piloting function continues to search a safe landing site but dummy, until the coarse landing point enters the camera FOV. No guidance decision is made based on dummy landing sites. Referring to Figure 14, the coarse landing region will be the region (red circles) closest to the target landing point (cross). The red line is the semi-major axis of the landing ellipse. The line of average squares (red circles) above the crossed location will be discarded. The new landing point is the centre of the landing ellipse. The landing ellipse function is under development.

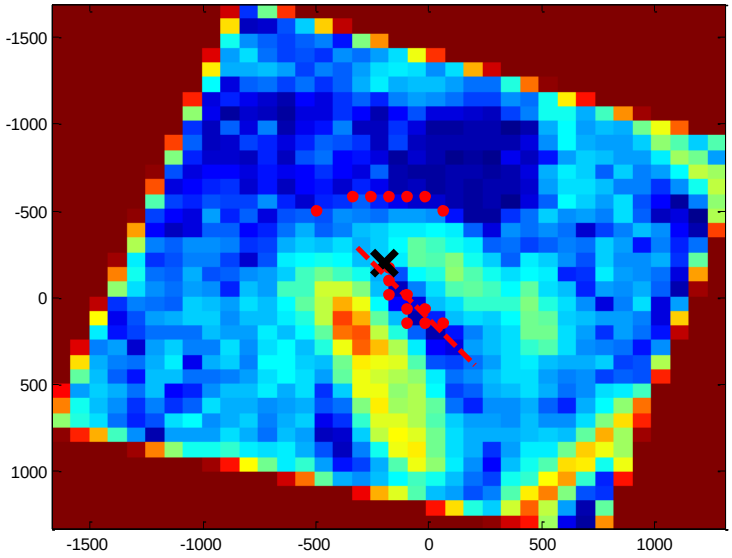

Figure 14. **Average square map with the target landing point and the closest safe squares**

Propellant ellipse P is defined as the region which is accessible given the current propellant with some margin. The equation of the ellipse is the same as (5.1). A landing site within the propellant ellipse is said to be reachable.

Average square algorithm given the map resolution or pixel width $w(h)$ takes the average of the safety level of macro-pixels of the hazard map inside the AS width.

$$\begin{aligned} s(k,l) &= \frac{1}{N(k,l)} \sum_{i,j \in \mathrm{S}(k,l)} s(i,j), N(k,l) > 0 \\ \mathrm{S}(k,l) &= \{ kw(h) \le x(i,j) < (k-1)w(h), lw(h) \le y(i,j) < (l-1)w(h) \} \end{aligned} \tag{5.3}$$

If the number of found macro-pixels $N(k,l) = 0$,which may be due to inclination of camera FOV, then interpolation is done from the previous contiguous AS with finite safety level

$$s(k,l) = \left(s(k \pm 1, l) + s(k, l \pm 1) \right) / 2, \; 0 \le s(k \pm 1, l) \le 1 \tag{5.4}$$

5.2.1 Phases Strategy

The best landing site is searched according to the following criteria:

1) The safety level of the landing site has to be lower than the threshold s_{max} (safe)
2) The landing site must be inside the propellant ellipse (reachable)
3) The landing site must be as close as possible to the target landing site (target).

To accomplish these objectives a strategy of phases is performed, Figure 15 shows the phases of the landing site algorithms. These phases are developed in order to have the piloting and guidance as incorporated as possible. The piloting exposed here is divided in five phases in which is assumed a single camera with a limited FOV and is taken into account the fact that the camera will lose sight of the target.

The piloting function has as main inputs

1) Projected hazard Maps (HM)
2) Time
3) Attitude (quaternion)
4) Centre of mass(COM) position (target frame of reference)
5) CoM Velocity vector,
6) Angular Velocity vector,
7) Nominal target landing ellipse,
8) Propellant ellipse is built by the piloting function,

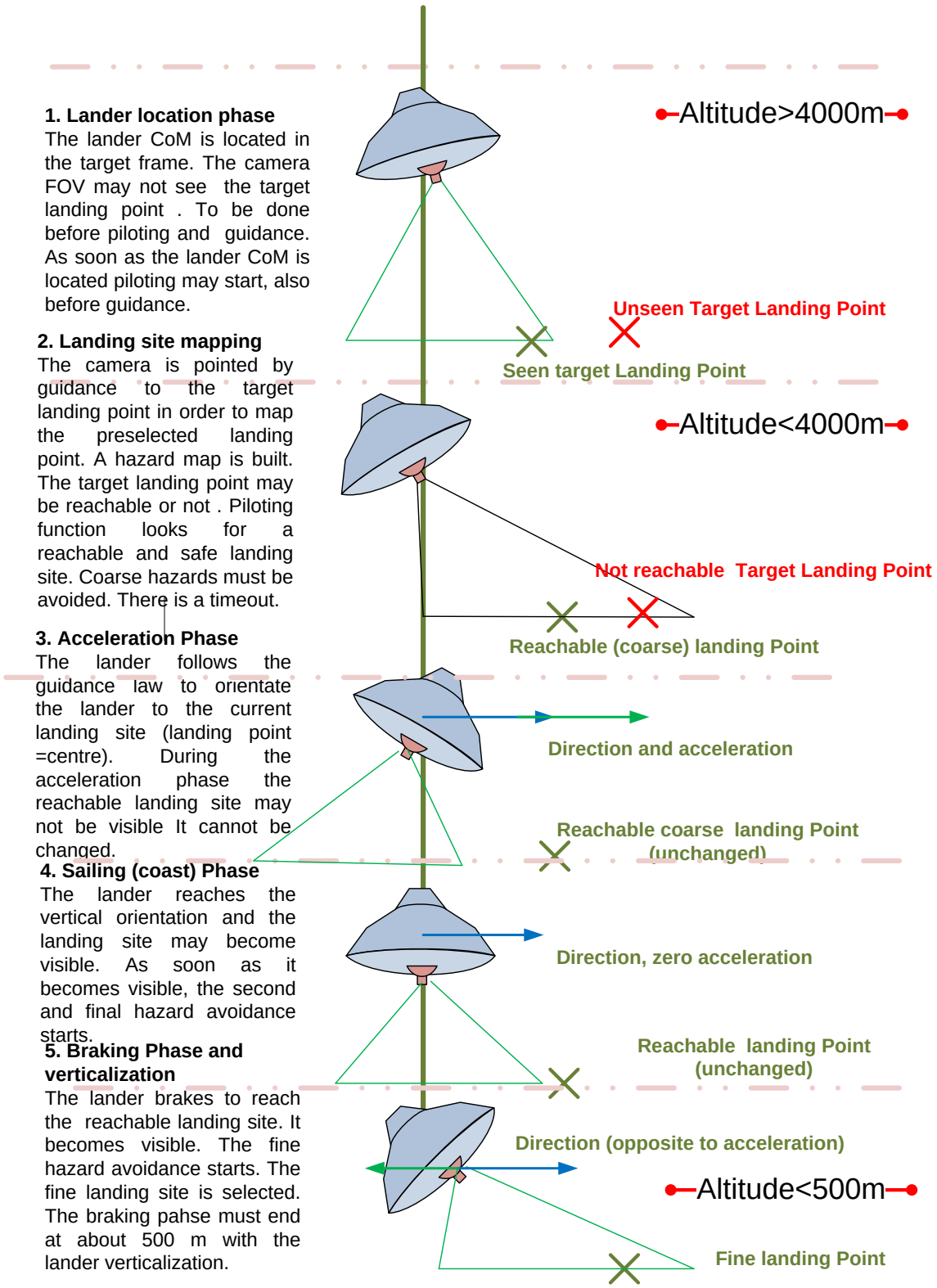

Figure 15. **Phases of the landing site algorithm.**

5.2.2 Phase 1- Lander location

The objective of this phase is to locate the lander in the local vertical local horizontal frame of reference and to start image acquisition, therefore obtain the first outputs of pre-guidance hazard maps, which is performed by navigation. The initial HMs are input to piloting and guidance. Lander location is performed as soon as navigation can establish the target frame and the lander CoM position, velocity and attitude (quaternion, angular rate) are detected in the frame. This phase is done during parachute descent, as soon as the front shield has been ejected. Altitude > 4000m. As soon as the above function is confirmed and hazard maps are received, piloting function may start by computing the propellant ellipse and selecting the best landing ellipse inside the available hazard maps (pre-guidance maps). The pre-guidance maps may not include the nominal target point, or exceptionally the target landing ellipse. To this end, the second phase is necessary. The phase duration is defined by navigation. If done during propulsion descent, time becomes critical.

5.2.3 Phase 2- landing site mapping

The phase aims to course map the region around the nominal target and find the current (coarse) landing ellipse. It is performed by piloting, guidance and navigation. As an exceptional case the landing point may be inaccessible because of limited lander tilt (45 degrees) or outside propellant ellipse. In both cases a new landing site is found close to nominal target by piloting function. Actually, in the former case the lander might be moved horizontally to approach the landing site.

Piloting imposes to guidance the lander orientation to centre the nominal landing site in the camera FOV. The actual guidance must be extended to implement attitude control alone. During lander orientation, to be fast, no map is obtained by camera. There are three possibilities, to be selected before guidance is actuated. (i) The target point can be viewed by the camera FOV at the end of the orientation (case 1). (ii) The landing point is outside the propellant ellipse (case 2) (Figure 16), but the landing point can be viewed by the camera. The lander is oriented to the landing point. (iii) (Case 3) The landing point cannot be viewed (large orientation angle). The largest orientation angle in the landing point is applied to the lander. The two last cases are due to control errors in the previous phases

(entry and parachute). The hazard map is elaborated to improve the current coarse landing site, free of coarse hazards. Hazards may occur at the nominal landing site because of location (knowledge) errors. The time duration must be short with a fast manoeuvre, without camera data, during slew orientation. Less than 4 s (400 m altitude).

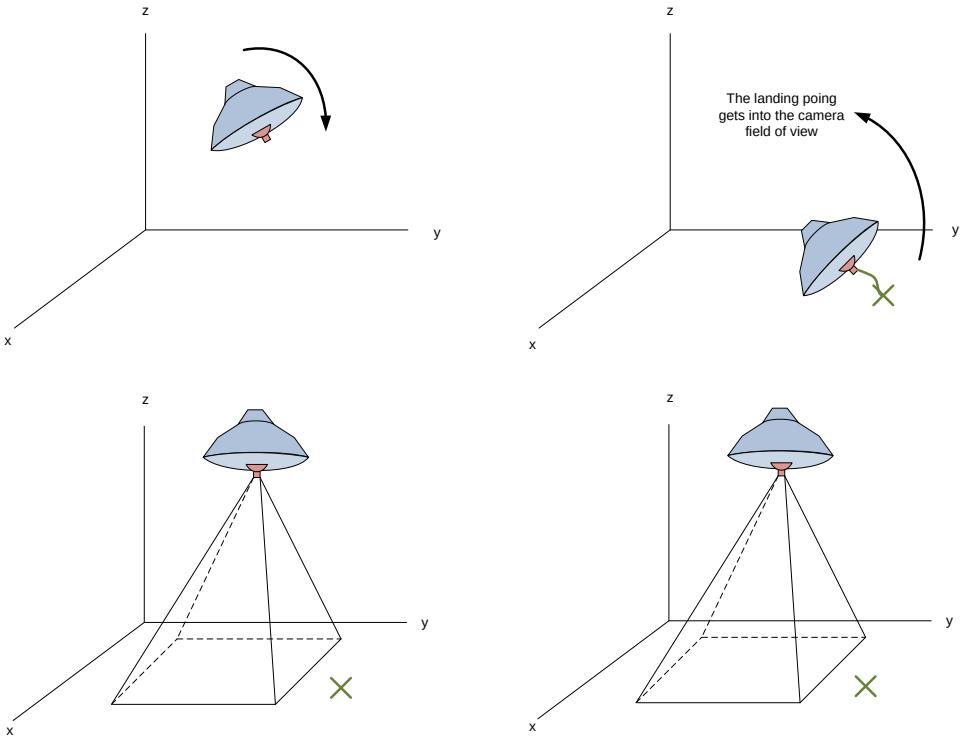

Figure 16. **Phase 2 left (case 1) right (case 2)**

5.2.4 Phase 3 – Acceleration

This phase aims to move toward the current lading site; it will require lander orientation or the opposite direction to accelerate with the axial thrusters. The guidance reorients the lander and accelerates toward the coarse landing site (Figure 17), during this phase, no camera map will be used. The lander will be oriented opposite to landing site and unless a second camera is available a single camera cannot enter the landing side in the FOV.

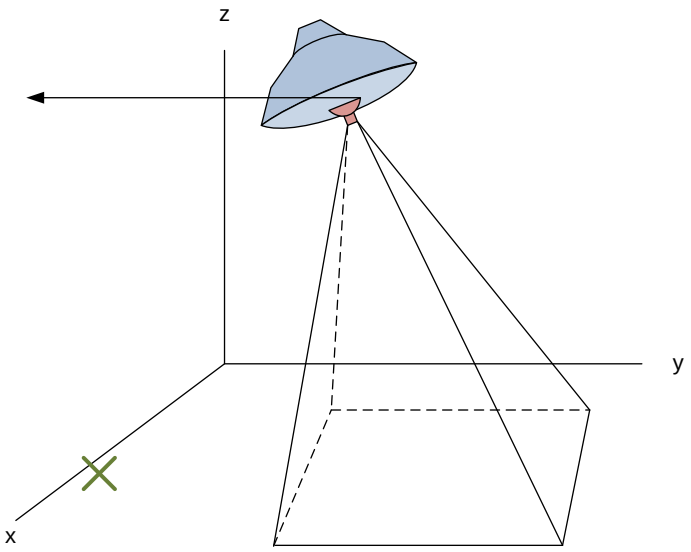

Figure 17. **Phase 3: acceleration toward the coarse landing site.**

5.2.5 Phase 4 – Coasting/braking and start for fine landing site.

This phase aims to move at constant velocity and then to brake toward the coarse landing site. The lander assumes the vertical orientation and then a tilt to brake (in the landing point direction), no horizontal acceleration or negative acceleration (Figure 18). In this way, the coarse landing site may enter the camera FOV. To this purpose the piloting starts again to elaborate hazard maps for finding a fine landing site (fine hazards).

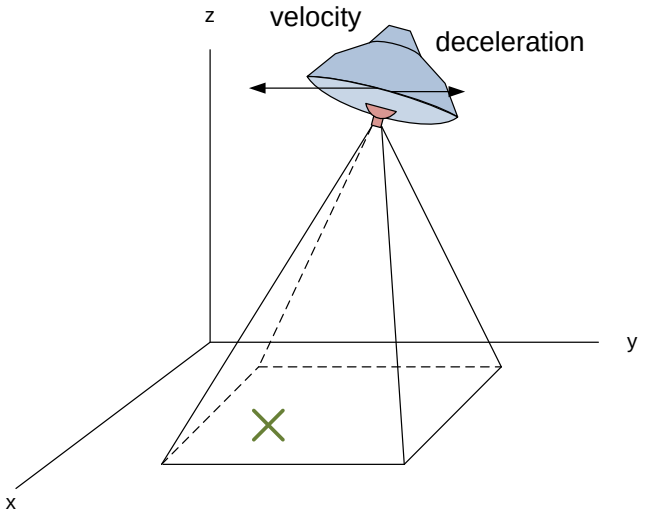

Figure 18. **Coasting/braking.**

The braking must be slow (low angular rate) to allow camera to map the coarse landing site. This phase should end at altitude of about 500 m to perform the final landing selection.

5.2.6 Phase 5 –Verticalization on the site and final small maneuvers.

This phase aims to orient the lander vertically on the fine landing site, ready to land. Final small manoeuvres can be made to avoid fine/small hazards. The lander assumes the vertical orientation in a fast way (no camera map) higher than 500 m altitude (< 20 m/s vertical velocity) at the end of horizontal braking. Final hazard map is elaborated to refine the final landing site. The final guidance is computed to approach the site (10 m altitude) at a vertical orientation.

Fine landing should end at about 500 m. After verticalization hazard map may continue to be elaborated, but the resolution being about 0.5 m, the map at 500m should be sufficient to plan the final guidance. Assuming 2 s after 500 m and 20 m/s of vertical velocity, the final guidance will start at about 400m. The final descent should be mostly vertical.

5.3 Hazard map processing to identify landing regions

The landing region must be defined in the initial phases, this process is performed by the piloting functions and guarantees the selection of candidates regions, these hazard map processing is made in order to choose large enough compatible regions with spacecraft footprint [10], [61]. Following treating each map is shown. Identification of the landing zones is made. The main selection criterion is to find the largest Circular zone (convex), that overcomes the safety level. Figure 19 shows an example where the safety level is given by a scale from zero to one. Zero indicates the highest safety (blue) and one the lowest safety (red). The hazard map coordinates are given in the local vertical local horizontal frame of reference.

A simulation example is performed based on HM in Figure 19. (Projected) Hazard map processing for piloting is subdivided in six steps as follows. Each step is explained in detail below.

1) Average square algorithm
2) Selection of the candidate *landing regions* through the safety threshold
3) Cluster selection.
4) Convexification of the selected cluster and search of the centre
5) Generation of landing ellipse
6) Restriction to the propellant ellipse

5.4 Average square algorithm

Typical Hazard maps are use in this step [18], the average square algorithm described in the section 5.2 is applied, the original HM is exposed in Figure 19, and the simulation result is shown in Figure 20.

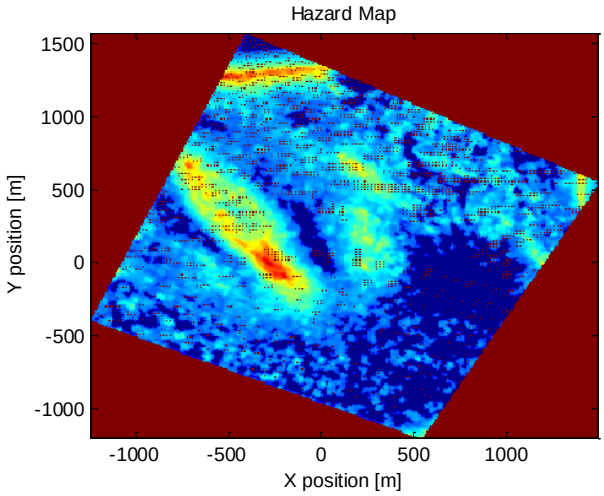

Figure 19. **Original HM**

Hazard map is compressed into the Average square map, this allows fast processing, and therefore computational load is reduced hence the real time implementation is feasible with a deterministic processing for Hazard maps algorithms.

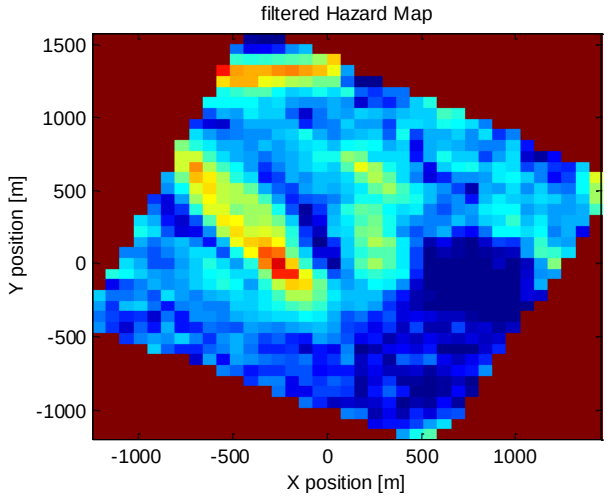

Figure 20. **Average square map from Figure 19**

5.5 Selection of the candidate landing pixels

The safety threshold is chosen. *Safe landing pixels* are selected to have a safety level below the threshold. Figure 21 shows the pixels found in the average square map of Figure 20. Green points correspond to safe landing pixels.

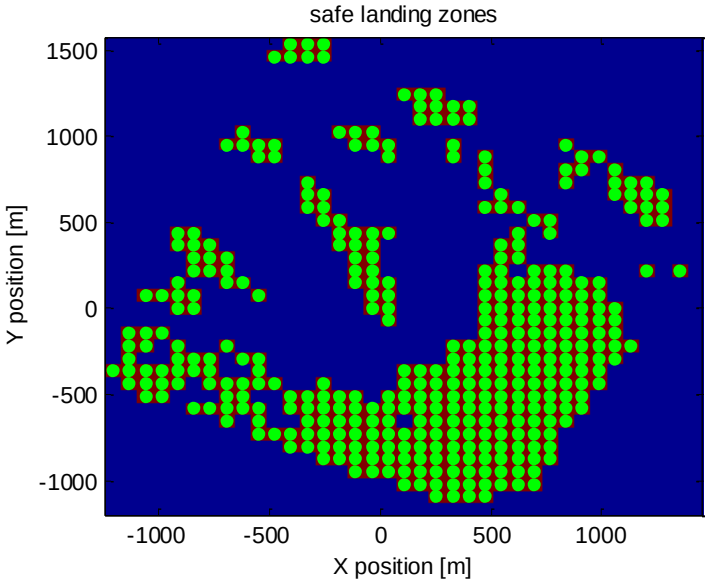

Figure 21. **Safe landing pixels of the ASM in**

As output of the algorithm step, *candidate landing pixels* are obtained. They are saved with their coordinates.

5.6 Cluster selection

Using the safe landing pixels, clustering and classification is made in order to find the *largest safe cluster*. To this end neighbouring pixels are grouped and clusters are determined.

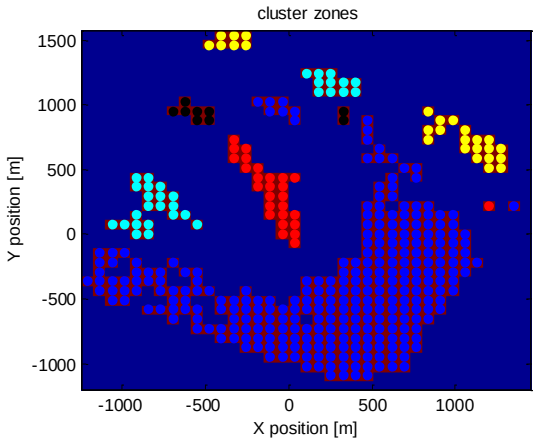

Figure 22. **Safe clusters of the safe landing pixels in**

5.7 Cluster convexification and centre determination

Obtained clusters may not be convex. A single cluster can be seen as a non convex polygon, and the process aims to find the largest convex sub-cluster. The convex sub-cluster must have a minimum area (landing footprint [10], [61]) that includes the uncertainty of the landing GNC. The following assumptions are made.

1) The landing site is selected to be convex and thus converted to an ellipse.
2) Each convex sub-cluster is defined by centre and diameter, prior of the ellipse conversion.
3) Each cluster includes at least one convex sub-cluster.
4) A convex sub-cluster is acceptable if the diameter is larger than the landing footprint.
5) Distance between neighbouring points is defined by the ASM.

To find the centre of the largest sub-cluster of a cluster, the following requirements are employed.

1) The centre must be inside of the cluster
2) The centre must be the farthest point from the cluster contour
3) There may be more than one sub-convex cluster in a single cluster.

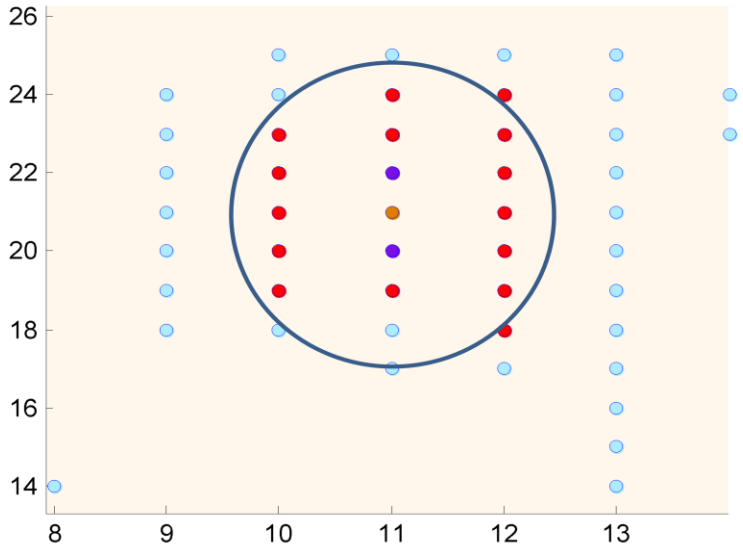

Figure 23. **Convexification algorithm example**

Each convex/non convex cluster is treated in order to find the centre of the sub-convex cluster. The algorithm is similar to [62]. The centre is found through a recursive algorithm that progressively removes the contour of the region until a single central point is achieved. The found point guarantees that is the centre of the maximum circle inscribed in the non-convex cluster. During the iteration process the non convex cluster may split. For this reason a further clustering process is added at each iteration. Therefore a single region may have more than one centre. After that, a set of regions defined by centre and diameter are obtained. A simulation example is shown in Figure 23 and the algorithm is explained graphically in Figure 24.

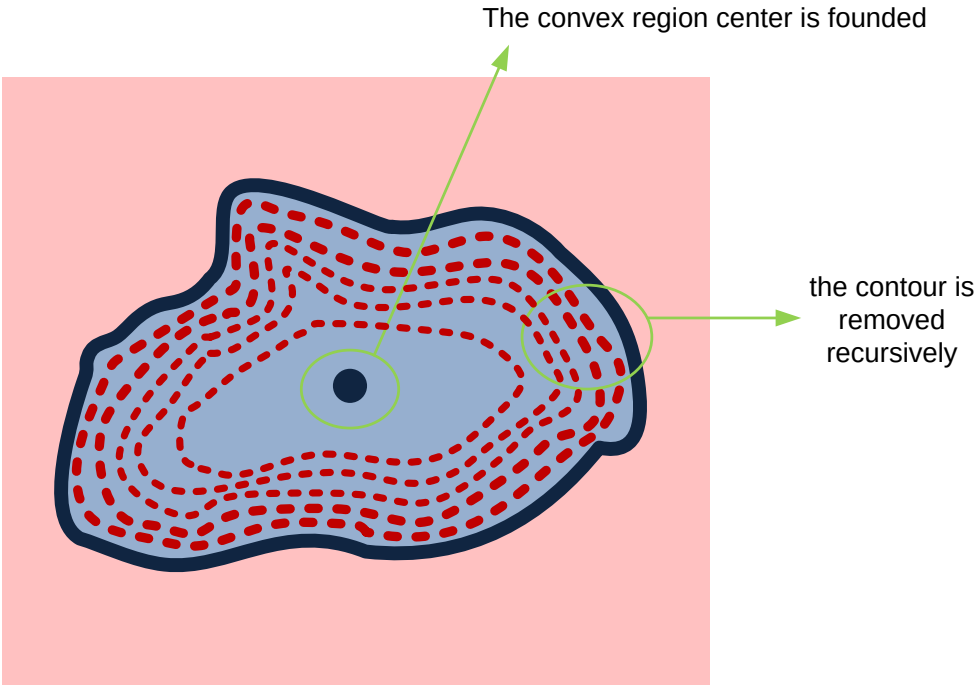

Figure 24. **Convexification algorithm and centre finding.**

5.8 Generation of the landing ellipse

Figure 24 shows the iteration process: the contour is shrunken, and at each step the algorithm generates smaller subclusters. The recursive breakpoint is defined by the sub-cluster being a single point. As soon as a centre has been found, the diameter is computed as the minimum distance between the original contour and the centre.

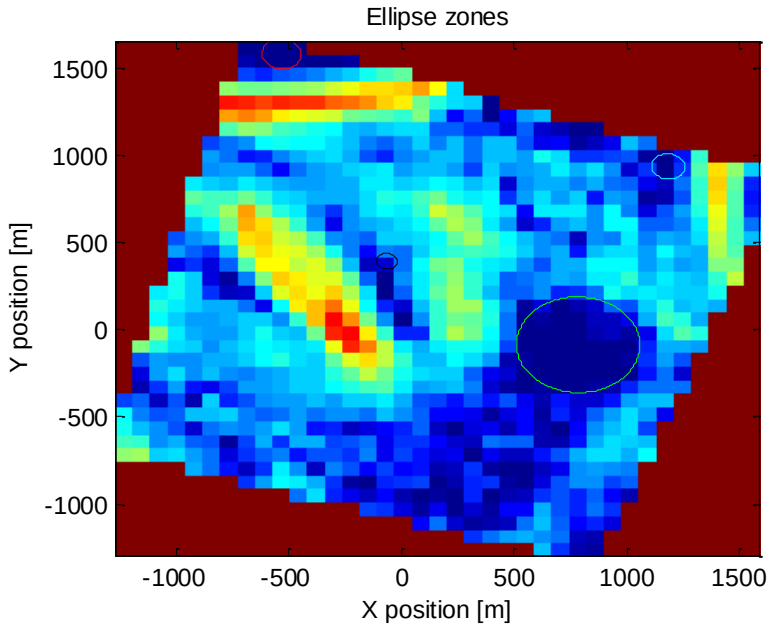

Figure 25. **Convex sub-clusters found.**

During the descent manoeuvre a set of hazard maps is available. The clusters are updated when a new hazard map is received. Fine hazards will be detected as soon as the height is reduced (some clusters can split or disappear). The field of view of the camera may lose some regions, in the descent the parameters of the regions (centres, diameters) are refined according to the new risks.

The result obtained from a typical landing path [11], [12] are shown in the figures below. The sub-clusters that intersect the propellant ellipse remain. For this simulation a propellant ellipse is assumed. Figure 26and Figure 27 shows the variations of the regions during the descent.

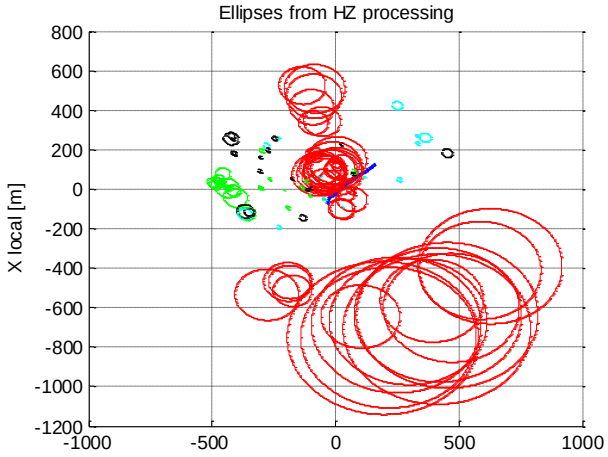

Figure 26. **Ellipses from Hazard Map processing (X-Y Plane)**

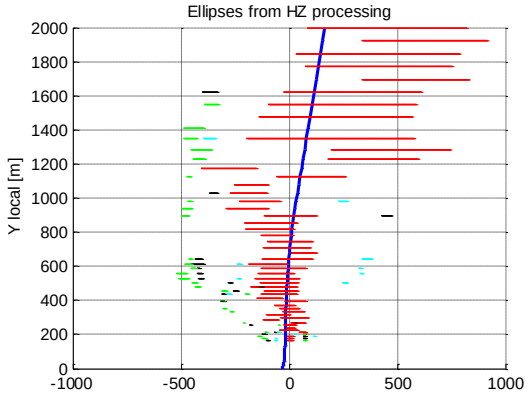

Figure 27. **Profile of ellipses at Hazard map height (X-Z Plane)**

5.9 Intersection with propellant ellipse

The sub-cluster that are complete or partial included in the propellant ellipse are candidate landing sites, Figure 28 shows the propellant ellipse (green) and the candidate landing sites.

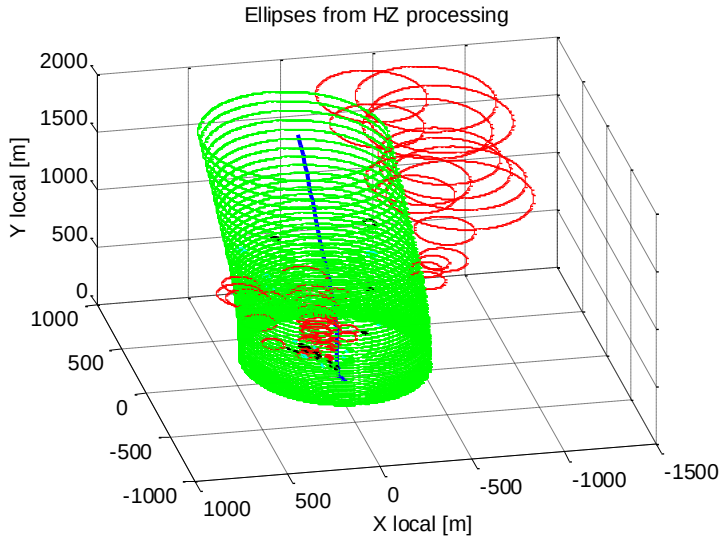

Figure 28. **landing path with safe regions found**

To summarize Figure 29 shows the scheme of the Piloting that is integrated with the guidance.

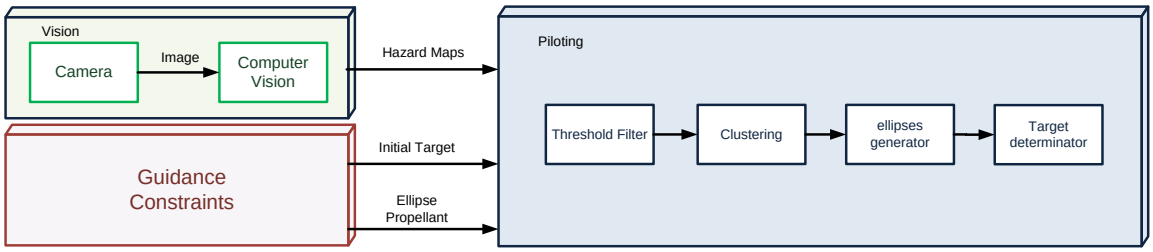

Figure 29. **Piloting Scheme**

5.10 Piloting – Guidance fusion

This section provides guidance and piloting are combined for the final stage of landing, the camera sensor is integrated to the simulator and maps are obtained through the orientation and position of the vehicle.

5.10.1 Piloting Simulation

In order to validate the piloting and guidance algorithms, an artificial mars surface was implemented (Figure 32) the projected hazard maps are obtained according to the lander attitude and position, to this aim a sensor of $36mm \times 24mm$ was used (Figure 30), focal length is determinate from the desired field of view.

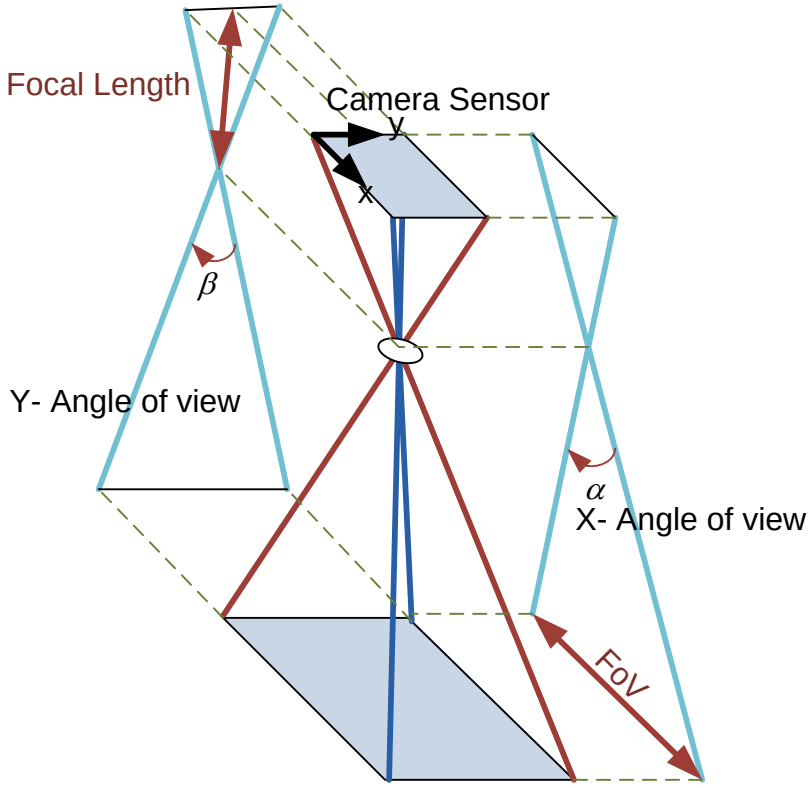

Figure 30. **Sensor Configuration**

The artificial surface region is generated through a random process passed by a filter. The sensor is divided in pixels each pixel generates a vector $\vec{s}_{bk}$ that is transformed and projected in the surface, a projection in the plane of the vectors $\vec{s}_{bk}$ is performed in order to obtain in-house hazard maps.

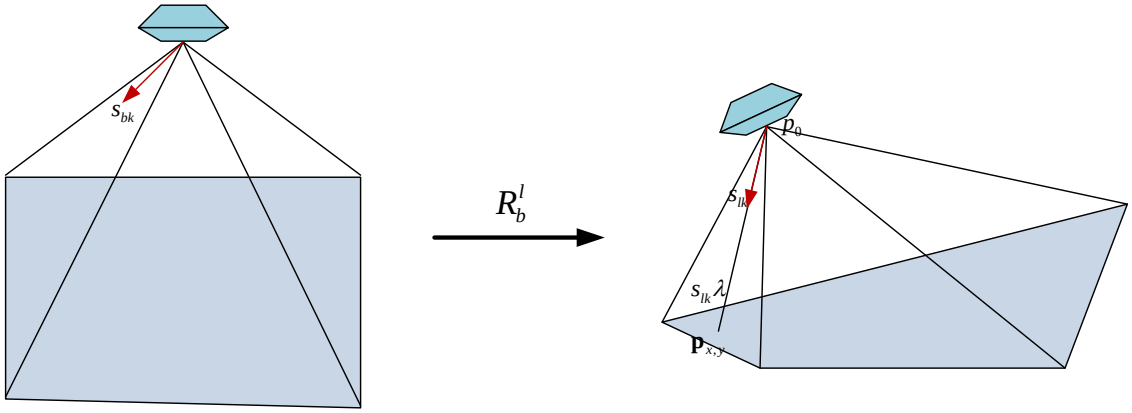

Figure 31. **FoV Projection**

$$s_{lk} = R_b^l s_{bk} \tag{5.5}$$

The interception of the vector with the target plane is called $\mathbf{p}_{x,y}$, the point is founded through the extension of the vector s_{lk}, the equation below shows the problem statement, where λ is the vector extension.

$$s_{lk}\lambda - p_o = \left(R_b^l s_{bk}\right)\lambda - p_o = \begin{bmatrix} p_x \\ p_y \\ 0 \end{bmatrix} = \mathbf{p}_{x,y} \tag{5.6}$$

$$\begin{bmatrix} \lambda \\ p_{11} \\ p_{12} \end{bmatrix} = \begin{bmatrix} 0 & 0 & \dfrac{1}{s_{lk3}} \\ -1 & 0 & \dfrac{s_{lk1}}{s_{lk3}} \\ 0 & -1 & \dfrac{s_{lk2}}{s_{lk3}} \end{bmatrix} \begin{bmatrix} p_{o1} \\ p_{o2} \\ p_{o3} \end{bmatrix} \tag{5.7}$$

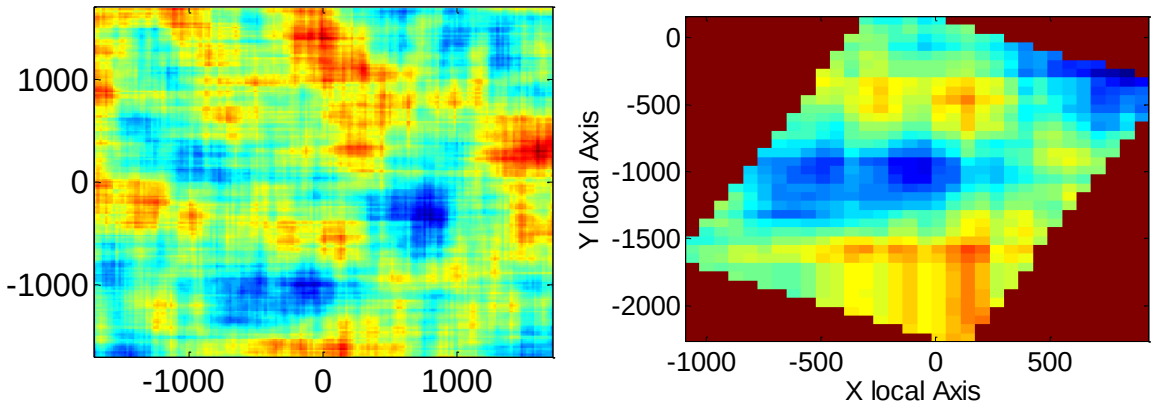

Figure 32. **Artificial Surface Region(left) in-house hazard map (right) (example h=1000m)**

5.10.2 Piloting-Guidance Simulations

This function was tested using in-house hazard maps since they are sensible to actual position and attitude. The following progressive runs are under development, using only guidance.

Table 1. Test 1

No	Altitude h[m]	Velocity (vertical) v [m/s]	Time to go tf	Lateral speed [m/s]	Tilt [rad]	Angular rate [rad/s]
0	1000	40	50	2	0.04	0.03
1	500	35	28	2	0.04	0.03
2	201	22	18	1	0.02	0.03

5.10.2.1 Height 1000m

Figure 33 shows in-house projected hazard map at a height of 1000 meters with the initial attitude, the algorithm obtain the ellipses that satisfied the safety level, for first tests the criteria used is to choose the largest region. The target selected is shown with a cross, each processed pixel has $64m \times 64m$ resolution.

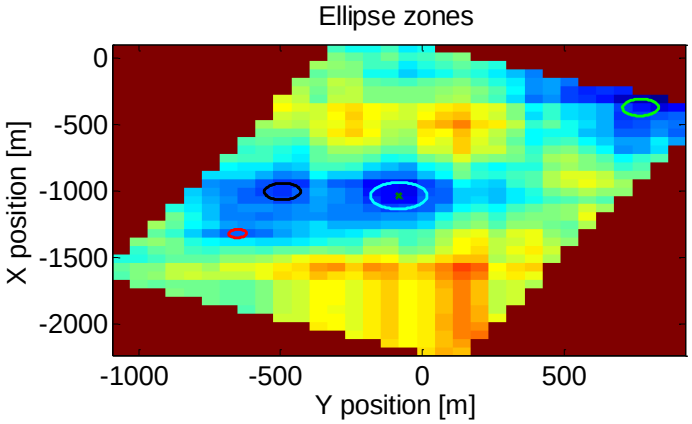

Figure 33. **Target selection based landing ellipse (Height 1000m)**

The position in the last phase of descent is shown in Figure 34, in this you can observe the correction to initial conditions and as it reaches the landing point selected by the hazard map processing.

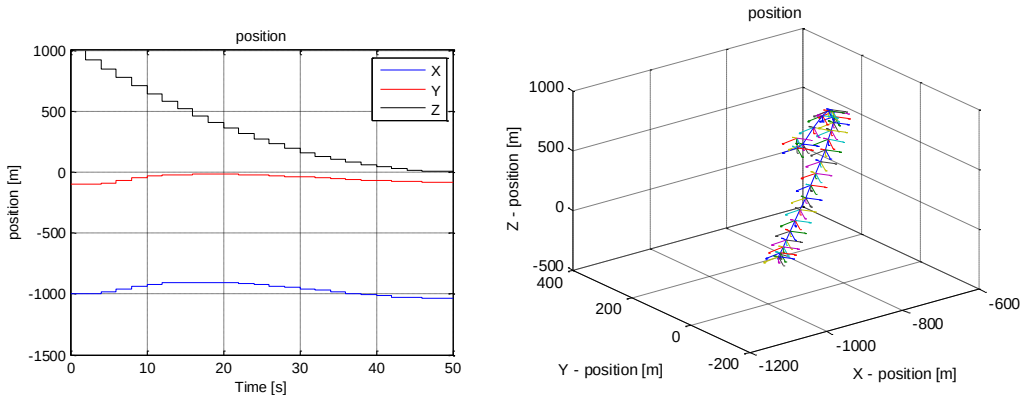

Figure 34. **Vehicle Position**

Figure 35 (right) shows how change the field of view as the vehicle descends, at each guidance step the simulator gets an in-house hazard maps and develops hazard map processing until 200m height, Figure 35 (left) shows the centres of the landing region obtained from each map.

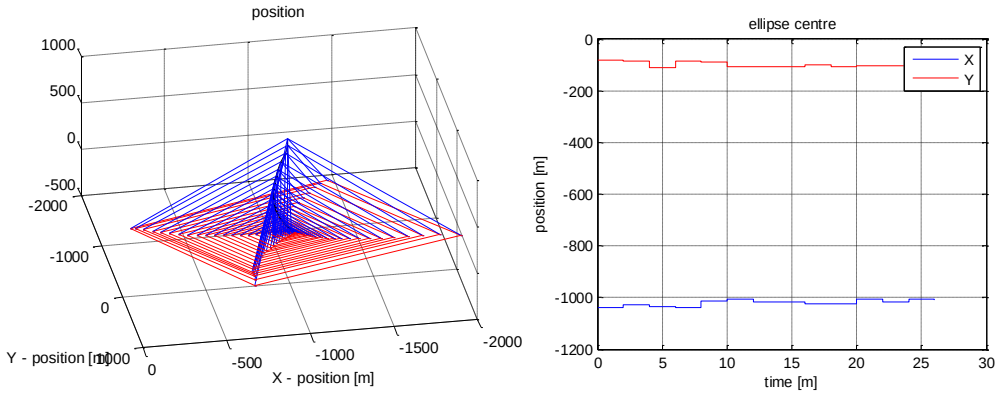

Figure 35. **3D Field of View (left) Landing Ellipse Centre (right)**

The tilt attitude guidance is shown in Figure 36, this attitude is based on the algorithms developed on [7], [11], [12], the graph is focused on the last phase of landing (<1000m) and how the zone is redefined.

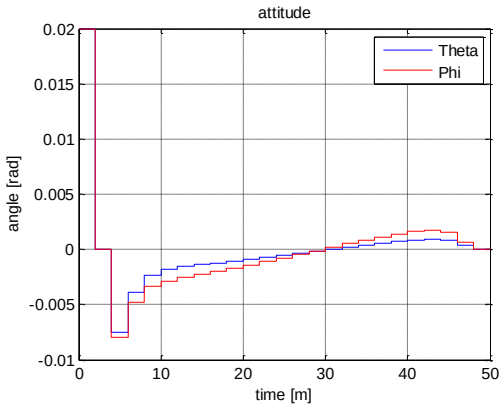

Figure 36. **Attitude Vehicle (1000m)**

5.10.2.2 Height 500m

The same results describe in are shown below for heights of 500 (Figure 37) and 201 meters (Figure 41). The target region is redefined with a new resolution and new risks that were not visible at higher altitudes appear.

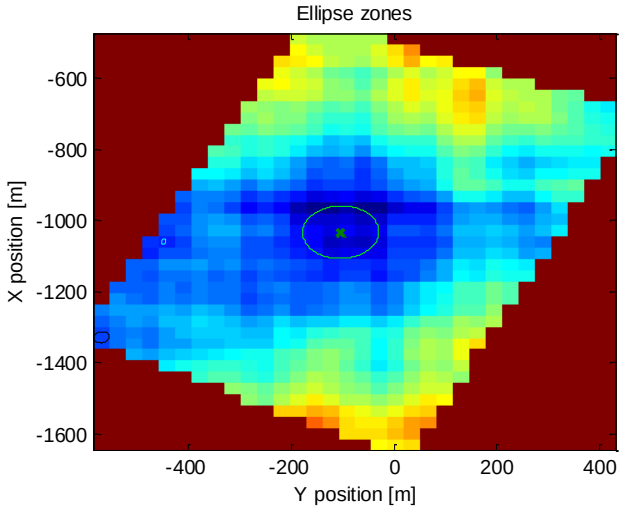

Figure 37. **Target selection based landing ellipse (Height 500m)**

The vector $\vec{s}_{bk}$ are shown in the position trajectory in Figure 38(right), they determinates the camera FoV.

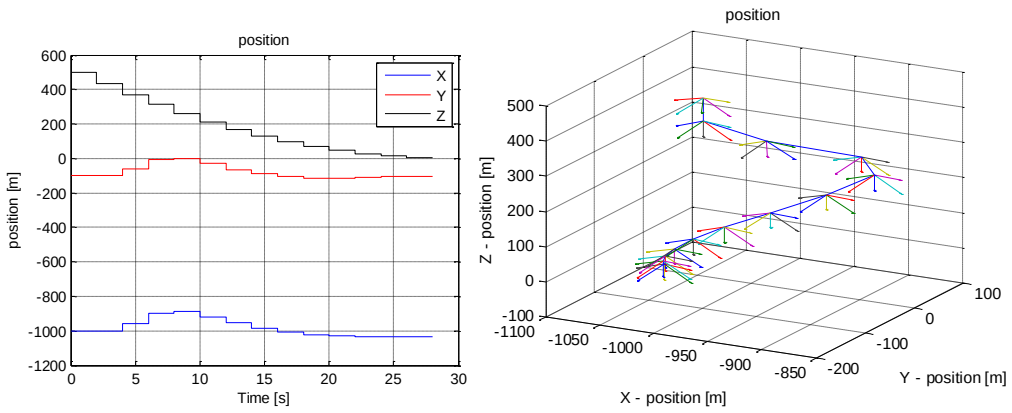

Figure 38. **Vehicle Position (Height 500m)**

The simulator obtains the hazard maps from the attitude and the camera FoV, this step is made for each photography.

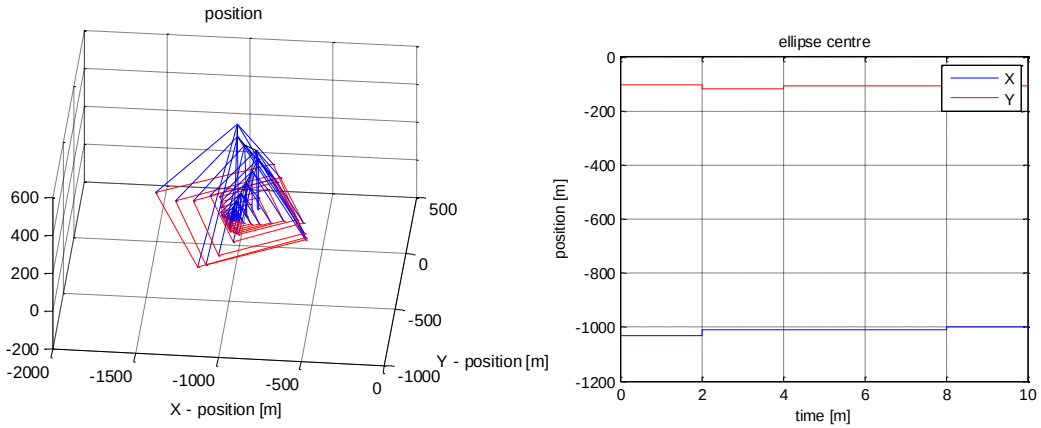

Figure 39. **3D Field of View (left) Landing Ellipse Centre (right)**

The horizontal guidance changes the tilt in manoeuvre beginning in order to modify the vehicle trajectory. The new images are taken at each photograph step and therefore the photo is taken in order to guarantee the camera constraints, Figure 39(left) shows the FoV at each photography step.

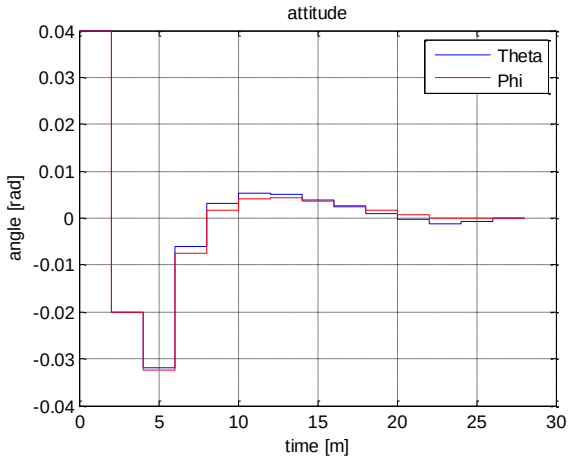

Figure 40. **Attitude Vehicle(Height 500m)**

5.10.2.3 Height 201m

Figure 41 shows the last map that is acquired by the algorithm, where each processed pixel has resolution $\approx 12m \times 12m$, the region is region is defined and refined to better resolution as the vehicle descends.

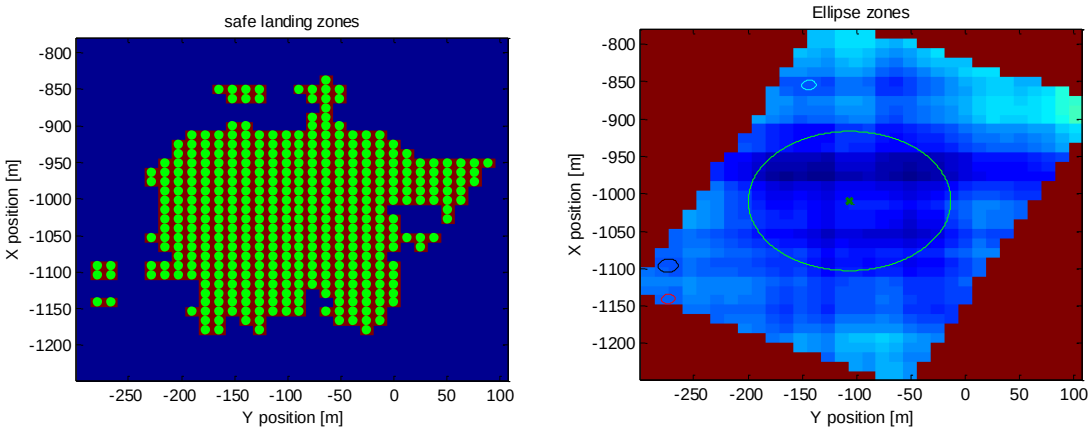

Figure 41. **Target selection based landing ellipse (Height 201m)**

The inclination shown in the graphic below is described as the behavior of the guide when a new hazard map is obtained.

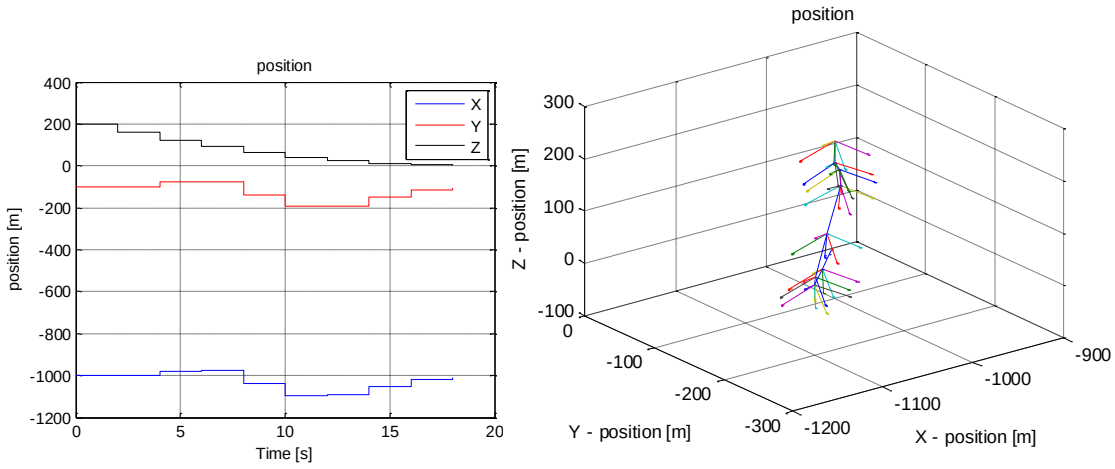

Figure 42. **Vehicle Position(Height 201m)**

In this section is shown the simulations results of the connection between piloting and guidance in the last phase of landing.

PART II BOREA QUADROTOR PROJECT

6. BOREA INTRODUCTION

The problem of on-ground testing guidance, navigation and control (GNC) algorithms for planetary accurate and safe landing can be approached through the flight of small quadrotors (< 2kg mass), suitable for indoor and outdoor operations. The literature is full of studies and experiments with such aerial vehicles [15], [41], [63]–[71] (sometimes referred to as aerial robots) aiming at very different exploration, commercial and education goals. Here we focus on the test of GNC algorithms for planetary landing. Simulated results will be presented as a baseline. The main difference of an on-Earth-flying quadrotor dynamics with respect to a generic planetary landing vehicle is analyzed, showing that a similitude can be formulated, capable of scaling down mass, geometry and trajectories to outdoor tests, and of compensating the different gravity acceleration. As a result, indoor tests look rather critical as they would require small quantization to quadrotor thrusts, in order to keep landing flight duration constant. Similitude to be very accurate needs a careful model of propeller dynamics and response. A further problem comes from emulating radar altimeter and velocimeter; both can be emulated by a GPS receiver but reliably only in outdoor tests. Indoor tests should require camera. Radar altimeters are massive. Altimeter is also emulated by Ultrasonic range sensors, they are used at touch-down. Initial alignment must be provided by some attitude sensor either magnetometers or external markers, or the accelerometer itself (on-ground). Subsequently, under a short flight time (< 100s), attitude, velocity and position can be obtained by gyro and accelerometer integration. Thus, essential sensor devices are assumed, namely IMU (accelerometers and gyros) and ultrasonic range sensor (conservative conditions). An outline of the guidance, navigation and control algorithms is included. Simulated runs are provided.

Preliminary to in-field tests a simulator has been built for testing and debugging the EMC code [6]. The control unit is built around the vehicle dynamics embedded model, which is a set of discrete time state equations of centre-of-mass (CoM) and attitude dynamics. It has been shown by [11], [12], [21] that the embedded model and the control strategies can be partitioned into horizontal, vertical and spin dynamics. Their cross-couplings are treated either as known or unknown disturbances to be estimated by state predictors (embedded

model plus noise estimator) and rejected by the control law. As a non-conventional feature, CoM horizontal dynamics and attitude dynamics (vehicle tilt, pitch and roll) constitute a unique dynamics. Assuming moderate tilt (acrobatic flight is not an immediate goal), the nonlinear dynamics from angular jerk to CoM acceleration is feedback linearized [42] to make available a fifth-order linear embedded model for each horizontal degree of freedom. Something similar has been done by [41].

The EMC design implies the development of the noise estimators, this navigation allows the control unit to estimate the variables accurately. In this research, a simple method to make state predictors that guarantee stability with minimum variance is presented. The horizontal noise estimator implements a decoupled strategy that allows the attitude variables to be estimated with a classical state predictor, this prediction is connected with the motion noise estimator through a nonlinear gain also predicted. In addition the horizontal navigation is the same for the vertical and horizontal position.

The vertical guidance uses a standard minimum control strategy [72]. The horizontal guidance splits in two identical models due to the feedback linearization in chapter 4. The strategy used is a polynomial strategy implemented with a two-phase state machine. The guidance avoids abrupt changes in the states, and thus the attitude variables are not forced, this feature is considered important to the stability of the quadrotor vehicles. The polynomial guidance also is adaptive to the measurements, and allows the reference states to be updated before a new maneuver.

The vertical control law is made with a classical state feedback. The horizontal case uses a state feedback with variable gains that are computed at each control step. Navigation is implemented using the embedded model and noise estimator structure typical of Embedded Model Control. The details presented are just a summary.

This research intends to develop the Borea quadrotor simulator. Specifically the simulator will include kinematic, dynamic, dispatching and will be able to determine the attitude and dynamics in the local and body frames of reference. The first contribution of this part of the

project is the development of a simulator of the quadrotor Borea, capable of describing the dynamics and kinematics of the vehicle.

In order to test the simulator a guidance, navigation and control strategy are implemented, this will follow the EMC scheme. This implies the development of the embedded models for position dynamics of the quadrotor. The embedded model is partitioned in horizontal and vertical dynamics and thus defining the strategy for the following procedures, the horizontal embedded model will include the attitude through a feedback linearization, this horizontal model reduce the complexity and transforms the quadrotor non linear model into a simple linear model with four controllable states no limited to small angles.

The EMC design implies the development of the noise estimators. This navigation allows the control unit to estimate the variables accurately. In this research, a simple method to make state predictors that guarantee stability with minimum variance is presented. The horizontal noise estimator implements a decoupled strategy that allows the attitude variables to be estimated with a classical state predictor, this prediction is connected with the motion noise estimator through a non linear gain also predicted and finally the horizontal navigation is the same for the vertical and horizontal position.

The vertical guidance uses a two order bang-bang control strategy through five steps state machine. This generator uses vertical acceleration and velocity bounds defined previously. The guidance includes one final state to correct discrete discrepancies, this guarantees to reach the correct position. The horizontal guidance is divided in two identical axes models due to the feedback linearization. The strategy used in this model is a polynomial strategy implemented in a two phases state machine.

The vertical control law is made with a classical state feedback. The horizontal case uses an adaptive state feedback with gains calculated and set at each control step; this implementation respects the linearization and guarantees a good performance with very low tracking errors.

6.1 Unmanned aerial vehicles UAV

The traditional aerial vehicles are commanded by an expert pilot onboard the aircraft. On the other hand, Unmanned Aerial Vehicles (UAV) are operated remotely or fly semi-autonomously based on algorithms preloaded on computers on board. They have been used for many applications; remote sensing [73]–[75], reconnaissance [76], scientific research [21], [22], [77], [78], civilian aerial-imaging [79], [80], aerial surveillance [81], [82], military [83], [84], inter alia. The most challenging problem of semi-autonomous algorithms is mainly to develop the GNC.

6.2 Quadrotors

The quadrotor vehicle has been previously analyzed due to their special features [77], [85], [86]. From the control point of view it is a non lineal plant with four driving forces produced by propellers. The quadrotor is composed by 4 propellers, in square distribution and interconnected with a rigid body composed typically by arms; the geometric centre is called C_p and is shown in Figure 43.

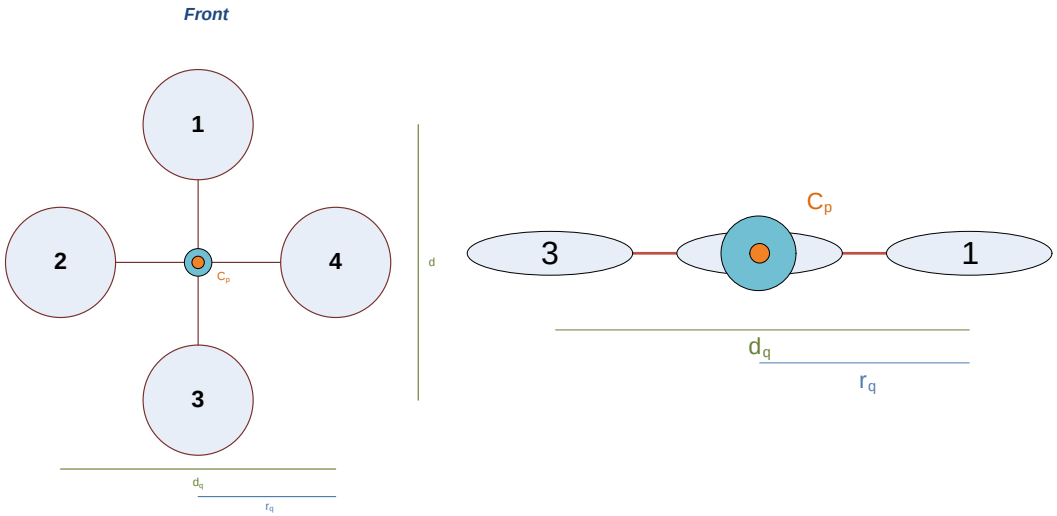

Figure 43. **Quadrocopter Left: Up view Right: Side Views**

The propellers are the actuators of the quadrotor, they transmit energy from rotational motion to the air, this effect produces a thrust command [24], [87], [88]. This topic is studied further in section 7.3.

7. QUADROTOR BOREA

The Borea quadrotor project [21]–[26] is developed by the Space and Precision Automatics (SPA) group from Politecnico di Torino. It aims to test Guidance, Navigation and Control (GNC) algorithms based on Embedded Model Control (EMC) theory. This consists in an Unmanned aerial vehicle (UAV), as is explained in section 6.1.

The quadrotor has important similarities with a precedent aircraft space vehicles in the landing propulsive phase, therefore this feature allows testing not only GNC algorithms for Earth flight but also for space applications, mainly because on-ground testing algorithms for accurate and safe planetary landing can be approached through the flight of small quadrotors[22]. A similitude between the spacecraft lander and quadrotor is analyzed and tested on chapter 12. Up to this date, the GNC algorithms for purposes of landing have been tested only in stochastic simulations.

7.1 Borea Quadrotor Modeling

7.1.1 Frames of references

As in section 2.3 the quadrocopter is located in a set of frame of references, additional references are added in order to analyze the thrust provide by propellers. This section describes the nominal frames of reference.

7.1.1.1 Inertial Frame of reference

The origin is located in Earth COM o_i, the axis x_i is located in the equator plane in direction of Greenwich meridian . The axis y_i is located in the equator plane orthogonal of x_i axis. The axis z_i is defined orthogonal to the previous axes $\vec{k}_i = \vec{i}_i \times \vec{j}_i$, the equation bellow summarize the frame.

$$R_i = \left\{ o_i, \vec{i}_i, \vec{j}_i, \vec{k}_i \right\}. \tag{7.1}$$

7.1.1.2 Local Frame of reference

All quadrotor requirements are referred to the local frame of reference. The following assumptions are applied;

4) The origin o_l is rigidly connected to the initial point target
5) the land is assumed plane for short displacements (<200m)
6) the z_l-axis is normal to the land plane
7) the x_l-axis and z_l-axis are assumed two orthogonal vectors in the land plane.

The vertical axis z_l is defined opposite to the gravity in the same direction of the Zenith

$$\vec{k}_i = -\frac{\vec{g}}{g}. \tag{7.2}$$

The axis x_l is located in the plane orthogonal to the axis z_l in the same direction to the Earth north. The axis y_l is defined as

$$\vec{j}_l = \vec{k}_l \times \vec{i}_l \tag{7.3}$$

and

$$R_l = \left\{o_l, \vec{i}_l, \vec{j}_l, \vec{k}_l\right\}. \tag{7.4}$$

Figure 44 shows the inertial frame of reference graphic scheme. A transformation between the inertial frame of reference and the local frame of reference is introduced. Figure 44 shows the transformation R_l^i that is defined by two rotations with the latitude ϕ_L and longitude λ_L angles, an exchange of the axis and signal inversion of an axis

$$R_i^l = \begin{bmatrix} 0 & 0 & 1 \\ 0 & -1 & 0 \\ 1 & 0 & 0 \end{bmatrix} Y\left(\phi_L\right) Z\left(-\lambda_L\right) \tag{7.5}$$

.

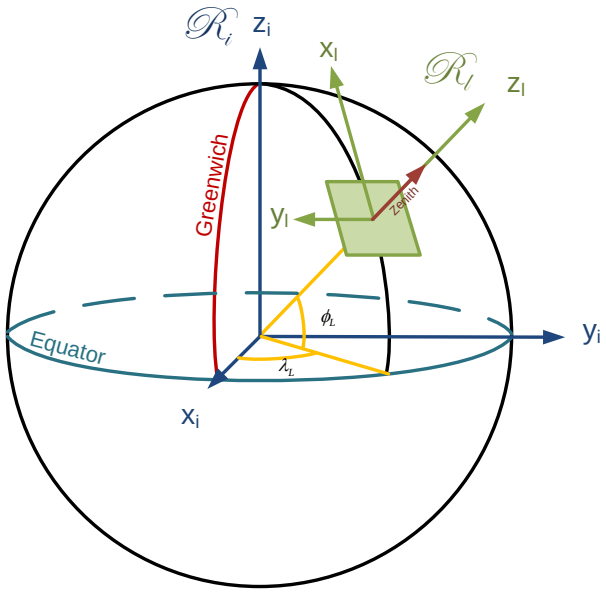

Figure 44. **Inertial and Local frames of references**

7.1.1.3 Body Frame of Reference

For the body frame of references the following assumptions are made,

1) all propeller centres lie in the same plane
2) the origin is the geometric centre of the propeller centres
3) the z_b-axis is normal to the plane in the direction of the thrust
4) the x_b-axis $\vec{i}_b$ is oriented as $\overrightarrow{A_3A_1}$
5) the y_b-axis is directed as $\overrightarrow{A_4A_2}$

The geometric reference is defined as,

$$R_b = \left\{O_b, \vec{i}_b, \vec{j}_b, \vec{k}_b\right\}. \tag{7.6}$$

The axes of this frame are

$$\vec{i}_b = \frac{\overrightarrow{A_3A_1}}{\left|\overrightarrow{A_3A_1}\right|}, \vec{j}_b = \frac{\overrightarrow{A_2A_4}}{\left|\overrightarrow{A_2A_4}\right|}, \vec{k}_b = \vec{i}_b \times \vec{j}_b \tag{7.7}$$

A vector represented in a geometric frame of reference can be transformed into an inertial frame of reference through the transformation matrix. The transformation matrix R_b^i is represented with three Euler angles ϕ_b, θ_b, ψ_b in 1-2-3 sequence. Figure 45 shows inertial and body frames of reference.

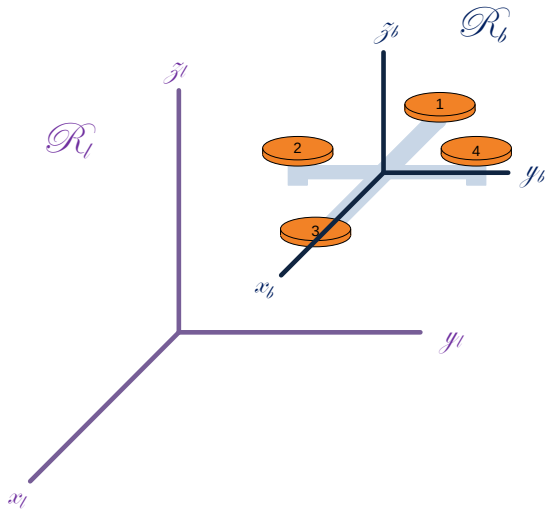

Figure 45. **Inertial and Body Frames of Reference**

7.1.1.4 Propeller frame of reference

For the propeller frame of references the following assumptions are made,

1) all propeller centres lie in the same plane
2) The frame assumes the arm orthogonal to the thrust vector.
3) the z-axis is normal to the plane in the direction of the thrust
4) The origin of each propeller frame of reference is located in the application point of the aerodynamic forces, it is the same as spin frame of reference

$$R_{pj} = \left\{ A_j, \vec{i}_{pj}, \vec{j}_{pj}, \vec{k}_{pj} \right\}, j = \{1,2,3,4\} \tag{7.8}$$

The axis $\vec{i}_{pj}$ is the direction of the arm $\vec{a}_j = \overrightarrow{C_p A_j}$ from the propeller centre to rotor centre, the reference of frame is shown in Figure 7 and defined as,

$$\vec{i}_{pj} = \frac{\vec{a}_j}{\left|\vec{a}_j\right|}, \vec{k}_{pj} = \frac{\vec{f}_{Tj}}{\left|\vec{f}_{Tj}\right|}, \vec{j}_{pj} = \vec{k}_{pj} \times \vec{i}_{pj} \tag{7.9}$$

This frame of reference is implemented in order to analyze the propeller aerodynamics effects and it is rigid connect with the geometric frame of reference. The angles between to translate a vector are fixed. Figure 46 shows the propeller frame of reference.

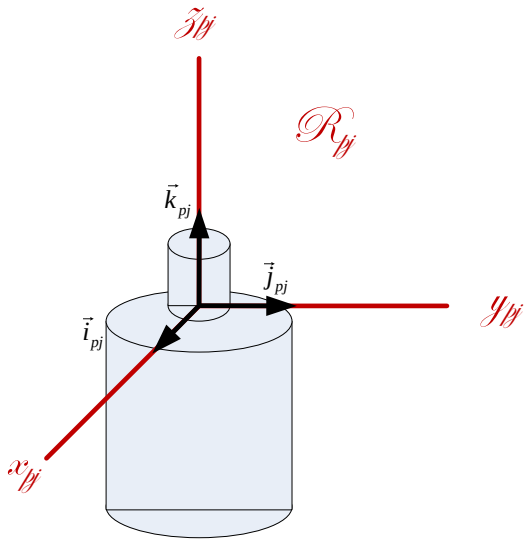

Figure 46. **Propeller frame of reference**

7.1.1.5 Spin Frames of Reference

For the spin frame of reference the following assumptions are made,

1) the $z_{sj,k}$ -axis is normal to the plane in the direction of the thrust
2) The axis $y_{sj,k}$ is the direction of the blade tip, equal to R_{bl} .
3) The axis $x_{sj,k}$ is in the same plane than the spin plane.

4) The origin of each spin frame of reference is located in the propeller application point of the aerodynamic forces.

$\vec{k}_{sj,k}$ is a normal vector to the plane formed by A_j points, the axis $\vec{k}_{sj,k}$ is defined as the direction of the nominal thrust $\vec{f}_{Tj}$.

$$R_{sj,k} = \left\{ A_j, \vec{i}_{sk,j}, \vec{j}_{sk,j}, \vec{k}_{sk,j} \right\}, j = \{1,2,3,4\}, k = \{1,..,n_b\} \tag{7.10}$$

The unitary vectors are defined as

$$\vec{j}_{sj,k} = \frac{\vec{t}_{j,k}}{\left|\vec{t}_{j,k}\right|}, \vec{k}_{sj,k} = \frac{\vec{f}_{Tj}}{\left|\vec{f}_{Tj}\right|}, \vec{i}_{sj,k} = \vec{j}_{sj,k} \times \vec{k}_{sj,k} \tag{7.11}$$

The following figure shows a representation of the spin frame of reference.

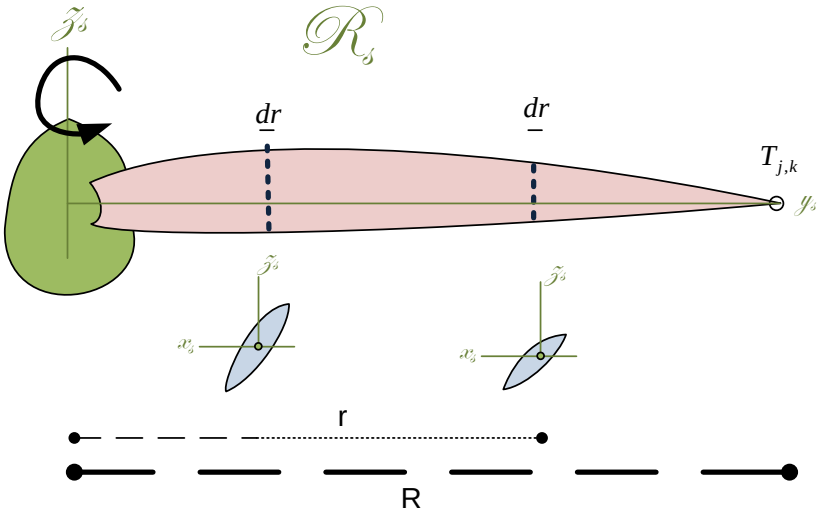

Figure 47. **Spin frame of reference**

The transformation matrix between the spin frame of reference and propeller frame of reference is defined as,

$$R_{sj}^{g}=\begin{bmatrix}\cos\theta_{gj} & 0 & \sin\theta_{gj}\\ 0 & 1 & 0\\ -\sin\theta_{gj} & 0 & \cos\theta_{gj}\end{bmatrix}\begin{bmatrix}1 & 0 & 0\\ 0 & \cos\phi_{gj} & -\sin\phi_{gj}\\ 0 & \sin\phi_{gj} & \cos\phi_{gj}\end{bmatrix}$$
$$R_{sj}^{g}=\begin{bmatrix}\cos\theta_{gj} & 0 & \sin\theta_{gj}\\ \sin\phi_{gj}\sin\theta_{gj} & \cos\phi_{gj} & -\sin\phi_{gj}\cos\theta_{gj}\\ -\cos\phi_{gj}\sin\theta_{gj} & \sin\phi_{gj} & \cos\phi_{gj}\cos\theta_{gj}\end{bmatrix} \quad (7.12)$$

The geometric reference frame and the spin reference frame have a common plane; to represent the translation is necessary only 2 angles. The angle θ_{gj} is related to the angular velocity of the propeller as follow equation,

$$\theta_{gj}=\Omega_j+\frac{2\pi}{3}(j-1) \quad (7.13)$$

7.1.1.6 Blade frames of reference

The blade frame of reference is implemented in order to analyze the aerodynamic force, it is not a unique frame but it is a set of frames of references which change along the blade and are connected to the spin frame of reference through the transversal-section.

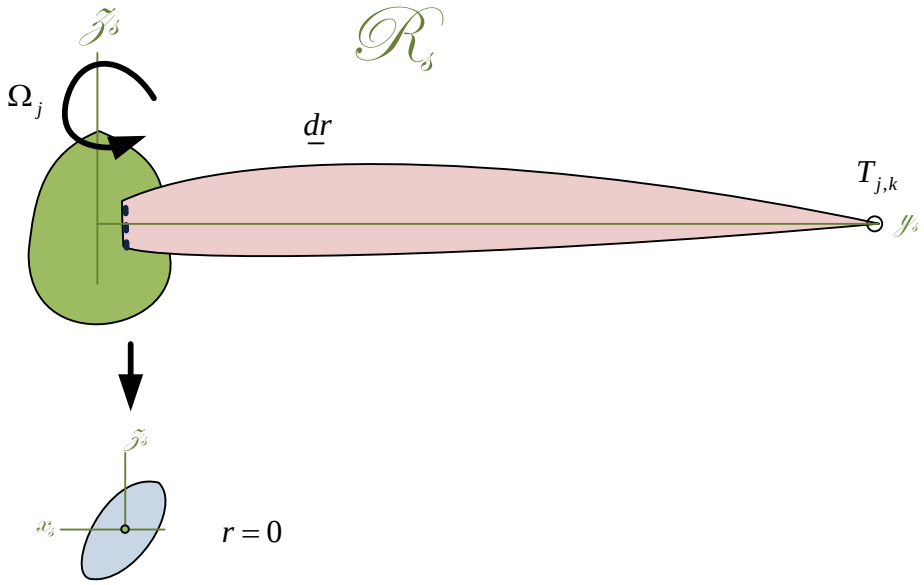

Figure 48. **Transversal section**

For the blade frame of reference the following assumptions are made,

1) The blades centres lie in the same plane.
2) The frame assumes the blade orthogonal to the thrust vector.
3) The blade reference frame is not unique for each blade, it is defined and change for each transversal section of the blade dr .
4) The origin O_k is located in propeller application point of the aerodynamic forces where the torques produced by these forces are equal to zero, it is for each transversal section.
5) The drag and lift blade coefficients are referred to this frame of reference

The blade frames of references are defined as

$$R_{blj,k} = \left\{ o_{k,j}, \vec{i}_{blj,k}, \vec{j}_{blj,k}, \vec{k}_{blj,k} \right\}, j = \{1,2,3,4\}, k = \{1,..,n_b\} \tag{7.14}$$

The axis $y_{blj,k}$ is the direction to the blade tip from the propeller centre $\vec{t}_{j,k} = \overrightarrow{A_j T_{j,k}}$, $\vec{i}_{blj,k}$ is in blade chord direction from the forces application point to the blade fin $\vec{g}_{j,k} = \overrightarrow{o_{k,j} G_{j,k}}$. The frame of reference is shown in Figure 48 and defined as,

$$\vec{j}_{blj,k} = \frac{\vec{t}_{j,k}}{\left|\vec{t}_{j,k}\right|}, \vec{i}_{blj,k} = \frac{\vec{g}_{j,k}}{\left|\vec{g}_{j,k}\right|}, \vec{k}_{blj,k} = \vec{i}_{blj,k} \times \vec{j}_{blj,k} \tag{7.15}$$

The translation between the spin frame of reference and the blade frame of reference is function of the pitch angle $\beta(r)$

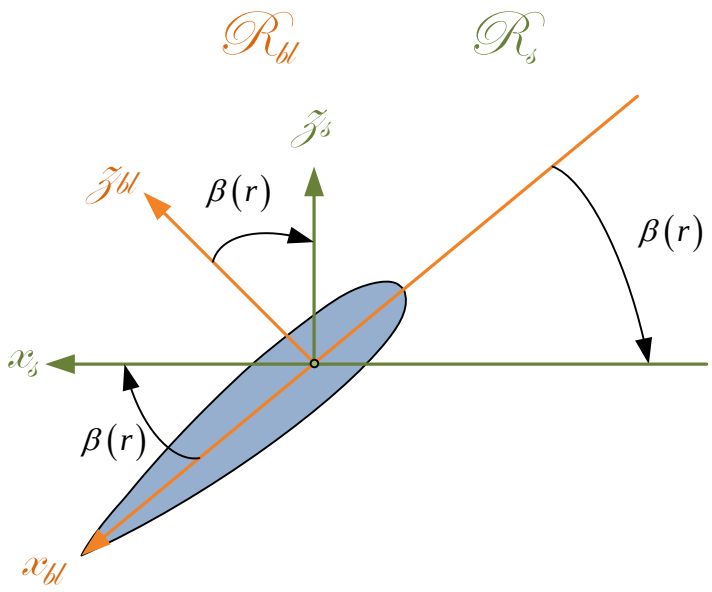

Figure 49. **Pitch angle**

The transformation matrix between the spin frame of reference and blade frame of reference is defined as,

$$R_{bl}^{s} = \begin{bmatrix} \cos\beta(r) & 0 & -\sin\beta(r) \\ 0 & 1 & 0 \\ \sin\beta(r) & 0 & \cos\beta(r) \end{bmatrix} \tag{7.16}$$

7.1.1.7 Wind frames of reference

The wind frame of reference share some properties with the blade frame of reference, it is unique for each transversal section dr and it takes the following assumptions

1) The relative wind velocity $\vec{v}_{R0\,j,k}$ is in the direction to the axis $\vec{x}_{wj,k}$.

2) and the axis $y_{wj,k}$ is the direction of the propeller centre from the blade tip opposite to R_{bl},

The wind frames of reference are defined as

$$R_{wj,k} = \left\{ o_{k,j}, \vec{i}_{wk,j}, \vec{j}_{wk,j}, \vec{k}_{wk,j} \right\}, j = \{1,2,3,4\}, k = \{1,..,n_b\} \tag{7.17}$$

$$\vec{j}_{wj,k} = \frac{\vec{t}_{j,k}}{\left|\vec{t}_{j,k}\right|}, \vec{i}_{wj,k} = \frac{\vec{v}_{R0\,j,k}}{\left|\vec{v}_{R0\,j,k}\right|}, \vec{k}_{wj,k} = \vec{i}_{wj,k} \times \vec{j}_{wj,k} \tag{7.18}$$

The transformation between the wind frame of reference and the blade frame of reference is denoted in equation (7.19).

$$R_w^{bl} = \begin{bmatrix} \cos\alpha_0 & 0 & \sin\alpha_0 \\ 0 & 1 & 0 \\ -\sin\alpha_0 & 0 & \cos\alpha_0 \end{bmatrix} \tag{7.19}$$

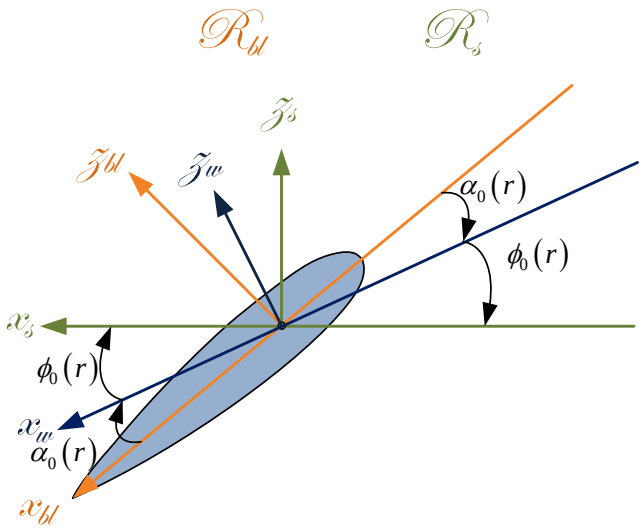

Figure 50. **Wind frames of reference**

All frames of references that act along the blade are depicted in Figure 51, these are necessary to understand the propeller operation, and allows to this research to develop a new novel method to modelling the propeller effects.

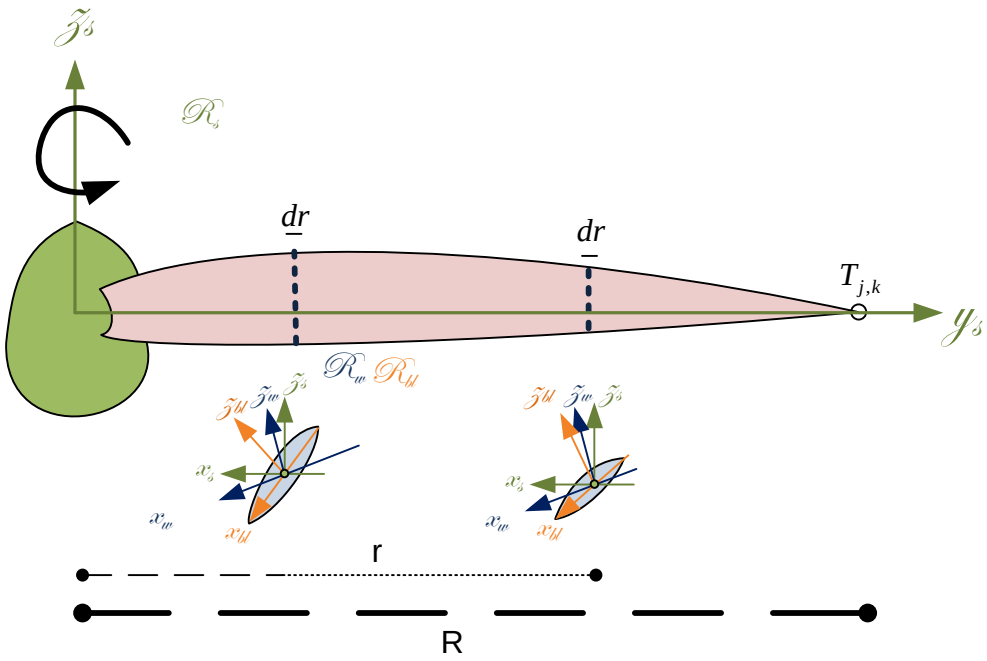

Figure 51. **References of frames on the propeller blades in two different transversal sections.**

7.2 BOREA SENSOR UNIT

Figure 52 depicts the main components of the Borea quadrotor [26],

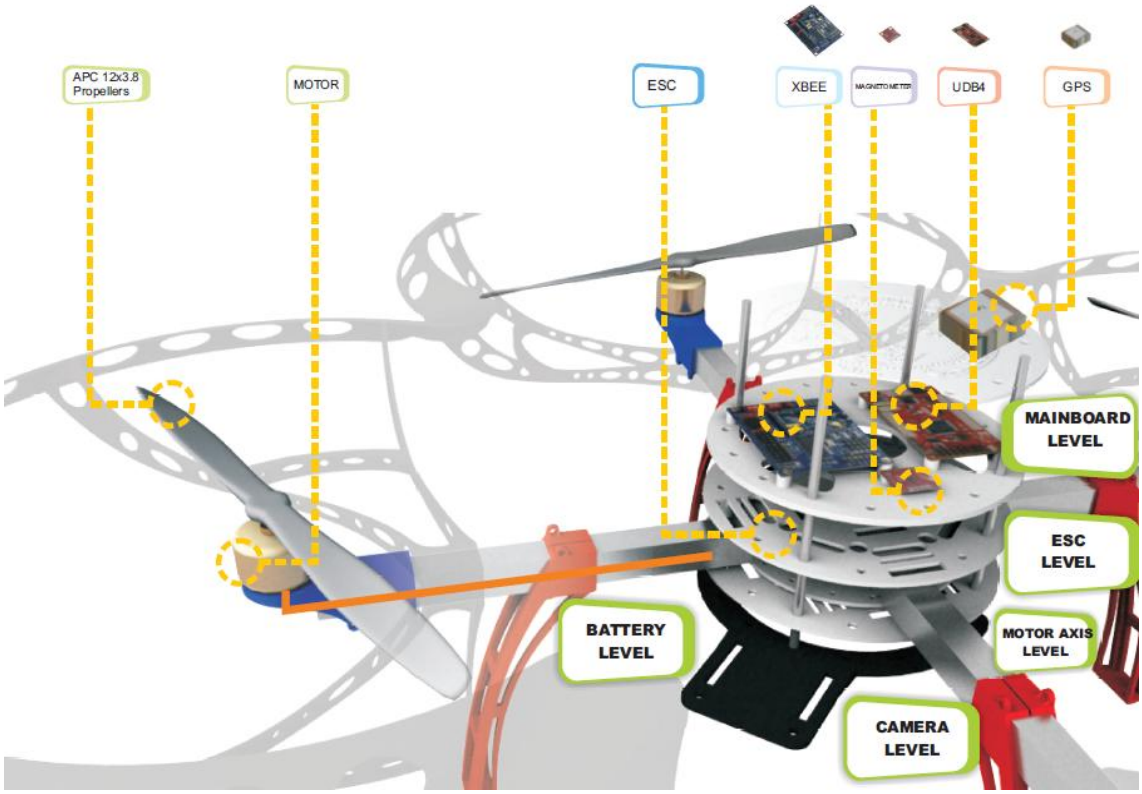

Figure 52. **Borea sensors and actuators**

7.2.1 measurement unit

The Borea quadrotor has a classical inertial measurement unit IMU, the accelerometer specifications are shown in Table 2, and a set of tests has been carried out on this sensor. The results are summarized in Table 3 , Figure 53 and Figure 54.

The accelerometer has an important role in the quadrotor navigation, in previous results [89], [90] show how can be used as an indirect measurement of the velocity through drag forces, and how can be include in a sensor fusion process with gyroscope as input of navigation. For the Borea quadrotor case a navigation has been divided in two levels, low

level navigation includes the sensor fusion, on the other hand the high level navigation is performed to get the variables and disturbances in the linear model studied in the first part of the dissertation.

Table 2. Accelerometer features

No	Variable	Symbol	Unit	Value	Comments
0	Zero-g	g	[V]	1.65	Output without acceleration
1	Sensitivity value	S	[mV][s^2]/[m]	81.549	The value of the sensitivity
2	Max-Output-Voltage	$V_{out\,\max}$	[V]	2.85	Vout=Vzero-g+ Sensitivity*1.5g
3	Min-Output-Voltage	$V_{out\,\min}$	[V]	0.45	Vout=Vzero-g- Sensitivity*1.5g
4	Max-Digital word-ADC		[LSB]	4096	12-bit then 2^12
5	Max-Digital word-ACC	$D_{\max acc}$	[LSB]	3537	Round(2.85 *4096/3.3)
6	Min-Digital word-ACC	$D_{\min acc}$	[LSB]	559	Round(2.85 *4096/3.3)
7	Range- Digital word-ACC	ΔD_{acc}	[LSB]	2978	
8	Rho voltage	$Rho_{acc}\|_{[V]}$	[V]/[LSB]	805.664 μ	3.3/2^12
9	Rho Acceleration	$Rho_{acc}\|_{[m/s$	[m]/[s^2]	9.87m	(3.3/2^12)/ 81.549m
10	Max-acceleration-1.5g	$a_{\max}$	[m]/[s^2]	14.715	g=9.81 m/s^2
11	Min-acceleration-1.5g	$a_{\min}$	[m]/[s^2]	-14.715	

All sensors on Borea quadrotor were characterized and tested [23], [25], the disturbances effects were analyzed and modelling. A low level navigation based on sensor fusion was implemented, the values of the position, velocity, attitude and angular rated are the outputs of the low level navigation and inputs of the high level navigation explained on chapter 10.

Table 3. Accelerometer Test						
No	TEST	CHANNEL	BIAS [V]	BIAS [M/S^2]	STANDARD DEVIATION[M/S^2]	COMMENTS
0	Accelerometer	X	0.3074	3.7692	0.0714	clean Environment test
1	Accelerometer	Y	0.0718	0.8804	0.0727	clean Environment test
2	Accelerometer	Z	0.3965	4.8622	0.0795	clean Environment test

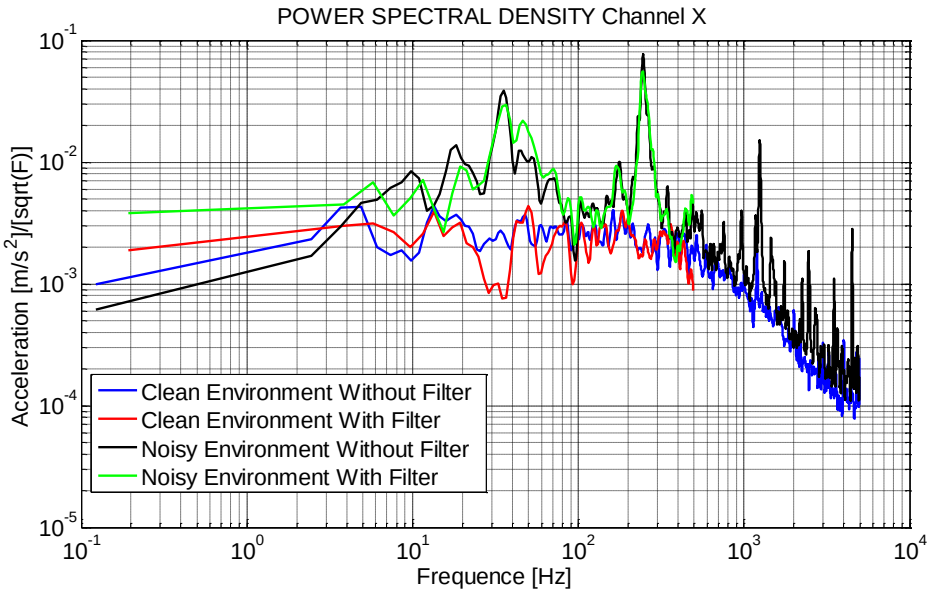

Figure 53. **Accelerometer PSD (Channel X)**

An example of a measurement performed by the accelerometer (axis x) is shown in Figure 54. The results were developed on a controlled laboratory (clean environment) and on a office desk (noisy environment), for the clean environment the sensors board was put into a "StableTop 450 Optical tabletops". It is an optical table enhanced with proprietary tuned and broadband dampers, where the outer skins are matched with the internal plate to minimize bowing due to thermal variations and cycling and to ensure the flatness of the tabletop is maintained. Moreover, the table has self-leveling vibration isolator that offers the highest performance of any pneumatic isolator. It also includes super damp isolators,

which provide total vertical and horizontal isolation with extremely low resonant frequency and constant height[1].

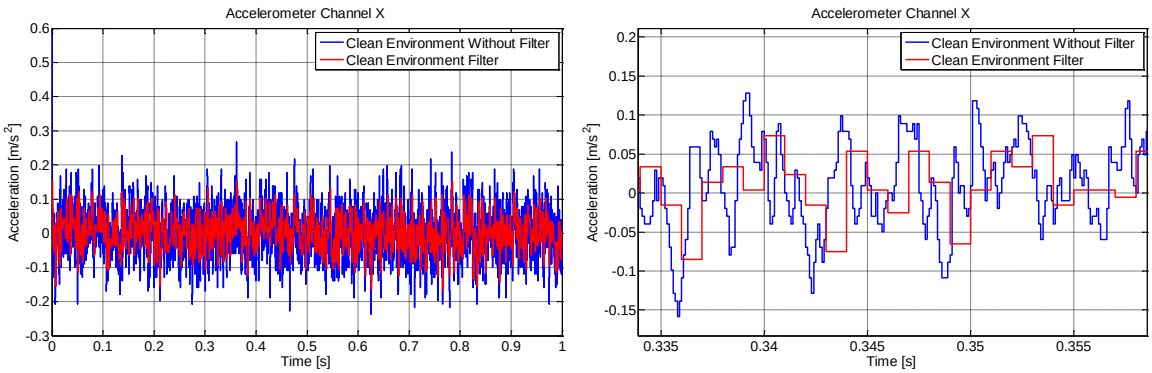

Figure 54. **Accelerometer measure (Channel X)**

Gyroscope, magnetometer, GPS and ultrasound were also characterized the Table 4,Table 5 and Table 6 show the main features that were verified and included on the simulator.

Table 4. Gyroscope features

No	Variable	Symbol	Unit	Value	Comments
1	Uscita zero-rate	V_{ZERO}	V	1.35	
2	Field Range	ω_{max}	rad/s	8.72	
3	Dynamic Range	S	mV/rad/s	114.7	
4	Tensione di uscita massima	V_{outmax}	V	2.35	$V_{outmax} = V_{zero} + S \cdot \omega_{max}$
5	Tensione di uscita minima	V_{outmin}	V	0.35	$V_{outmin} = V_{zero} - S \cdot \omega_{max}$
6	Massima word digitale	D_{max}		2917	$D_{max} = V_{outmax} / \sigma_{ADC}$
7	Minima word digitale	D_{min}		434	$D_{min} = V_{outmin} / \sigma_{ADC}$
8	Intervallo di quantizzazione	σ_{ADC}	µV	805.66	$\sigma_{ADC} = 3.3V / 2^{12}$
9	Risoluzione	σ_{GYR}	mrad/s	7.031	$\sigma_{GYR} = \sigma_{ADC} / S$

[1] The optical table was provided by Thales Alenia space.

Table 5. Magnetometer features

No	Variable	Symbol	Unit	Value	Comments
0	Field Range	g	[uT]	±800	
1	Dynamic Range	S	[uT]	±88	
2	Sensitivity	S	uT/Lsb	0.07299	In another unit is 1370Lsb/gauss
3	Noise floor		uT	0.2	
5	Measurement Period	Tcon	ms	20	Time from receiving command to data ready

Table 6. Magnetometer Parameters

No	TEST	CHANNEL	STANDARD DEVIATION[T]	AVERAGE[T]	AVERAGE[LSB]	COMMENTS
0	POSITIVE BIAS	X	2.0696e-007	1.1263e-004	1.5430e+003	
1	POSITIVE BIAS	Y	5.5958e-007	1.0586e-004	1.4503e+003	
2	POSITIVE BIAS	Z	2.8622e-007	1.1104e-004	1.5213e+003	

7.3 THRUST UNIT

A quadrotor vehicle has several differences with a planetary lander in a propulsive phase, one of the main differences is the propulsion system. In order to make a similitude, the thrusters used in planetary landing are replaced by propellers, therefore a complete study of the aerodynamics effects of the propeller was done.

The propeller analysis is a problem well known in literature, most of the studies use an approximation to the force is the square of the angular velocity [63], [69], in this research a novel method to develop a force model, which can be compute in real time is developed.

7.3.1 Propellers

The operating principle of the propeller is to transfer rotational energy in order to generate propulsion, to this aim the air mass is displaced through the blades and therefore a force is generated. From the physics point of view this force can be obtained mixing two different theories, first using the Bernoulli's principle that is called momentum theory, and the second one airfoil theory that is used to consider the blades, this is commonly called blade element theory. Each analysis gives and expression of the force, and finally both are combined to obtain the complete expression of the thrust $\vec{f}_T$ [24], [87], [88].

7.3.2 Momentum theory

This theory is focused on the pressure difference on the propellers. The propeller can be seen as a tube that displace atmospheric air mass trough the blades, the force is expressed in terms of this pressure differences. For this analysis the propeller frame of reference (section 7.1.1.4) is used and the following assumptions are considered,

- The propeller blades are despised.
- The propeller is a ideal disk infinitely slim.
- The air that goes through the propeller is seen as an incompressible fluid.
- The propeller does not have resistance with the air.
- The air changes the velocity magnitude and maintains the direction
- The force that impulses the air $\vec{f}_a$ is opposite to the propeller thrust $\vec{f}_T, \vec{f}_T = -\vec{f}_a$
- The forces are applied in the disk center. torques are not included in the momentum theory.

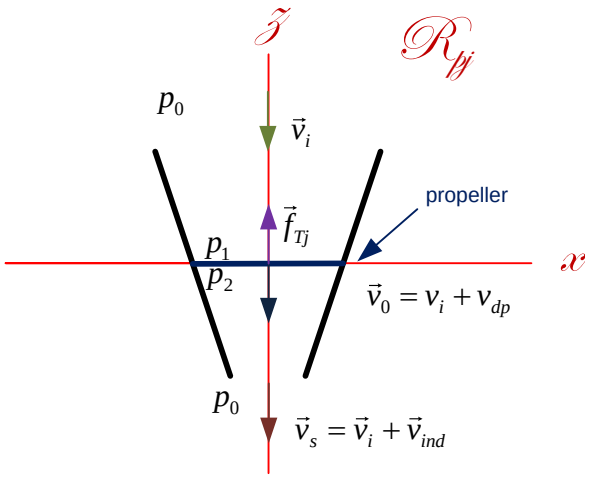

Figure 55. **Axial wind flow through the propeller**

Figure 55 shows a simple scheme of the pressures, flows and forces that are significant in the momentum theory, where wind velocity on top (initial) of the propeller is denoted as $\vec{v}_i$, wind velocity through the propeller is denoted as $\vec{v}_0$, wind in the bottom of the propeller is denoted as $\vec{v}_s$ which can be expressed as an increment $\vec{v}_{ind}$ from the initial wind velocity $\vec{v}_i$, i.e. . $\vec{v}_s = \vec{v}_i + \vec{v}_{ind}$. The pressure in the lower part of the propeller is denoted as P_2 and in the upper part is denoted P_1 and the atmospheric pressure is denoted P_0. Thrust $\vec{f}_T$ can be expressed in terms of the pressure difference on the disk as.

$$f_T = S(p_2 - p_1) = S\Delta p \tag{7.20}$$

On the other hand an additional result is obtained from air linear momentum $\vec{p}_d$. In the disk, the conditions are given by the atmospheric pressure P_0 and the initial air's velocity $\vec{v}_i$. The mass of air which pass through the disk is defined by the area S, the disk air's velocity $\vec{v}_0$ and the air's density ρ. The flow of mass through the disk is,

$$\vec{p}_d = m_a \vec{v} \tag{7.21}$$

$$\dot{m}_a = \rho S v_0 \tag{7.22}$$

Where m_a is the air mass. Thrust is expressed in terms of the velocities as,

$$\vec{f}_T = -\rho S v_0 \left(\vec{v}_i - \vec{v}_s\right), \Delta p = \rho v_0 \left(\vec{v}_i - \vec{v}_s\right) \tag{7.23}$$

The variation of the linear momentum is expressed as,

$$\vec{f}_T = \dot{\vec{p}}_d = \rho S v_0 \left(\vec{v}_i - \vec{v}_s\right) = -\rho S v_0 \vec{v}_{ind}, \tag{7.24}$$

The velocity v_{ind} is found through the fusion of the momentum theory and blade element theory, explained before.

$$v_{ind} = \sqrt{\frac{2 f_T}{S\rho} + v_i^{\,2}} - v_i \tag{7.25}$$

7.3.2.1 Efficiency analysis

The previously analysis was made based on several assumptions, therefore the actual efficiency is lower because:

3) Slipstream rotation
4) Blade profile drag
5) Non-Uniform flow
6) Compressibility effects
7) Propeller Blockage (Fuselage, Nacelle)

But in ideal cases is not possible use all generated power as required power, part of the energy is dissipated in the ideal propeller operation. The following procedure aims to find the analytical expression of the efficiency. The incompressible Bernulli equation can be applied only to constant energy flow, therefore this equation is applied in two stages, before,

$$\frac{\rho v_i^{\,2}}{2} + p_0 = \frac{\rho v_0^{\,2}}{2} + p_1 \tag{7.26}$$

And after,

$$\frac{\rho v_0^{\,2}}{2} + p_2 = \frac{\rho v_s^{\,2}}{2} + p_0 \tag{7.27}$$

Therefore the pressure difference can be seen as,

$$\Delta p = p_2 - p_1 = \frac{\rho}{2}\left(v_s^{\,2} - v_i^{\,2}\right) = \rho\left(v_i + v_{ind}/2\right)v_{ind} \tag{7.28}$$

From (7.23) and (7.28)

$$\left(v_s + v_i\right)/2 = v_0 \Rightarrow \left(v_i + v_{ind} + v_i\right)/2 = v_0 \tag{7.29}$$

The useful generated power P_u of the propeller is given by,

$$P_u = \vec{f}_T \cdot \vec{v}_i \tag{7.30}$$

The ideal propulsive efficiency or the Froude efficiency of a propulsive system is,

$$\eta_d = \frac{v_i}{v_0} \tag{7.31}$$

7.3.3 Simple Blade-Element Theory

The airfoil theory can be used in order to determine the components of thrust and torques that are produced by the blade. For this case the blade frame of reference and wind frame of reference are utilized in order to analyze each transversal section along the blade. For this analysis are used the following assumptions,

- $\vec{v}_b$ is the forward velocity and is located in the air plane
- the real blade velocity $\vec{v}_R$ takes into account the propeller advance velocity and The spin velocity

The effect of the air on the propeller dynamics is represented by the total aerodynamic force $\vec{f}_p$, this force is the result of the addition of each blade-aerodynamic force, where n_b is the number of blades.

$$\vec{f}_p = \sum \vec{f}_{bk}, k = \{1,..,n_b\} \tag{7.32}$$

The aerodynamic force is expressed in a spin frame of reference as,

$$\mathbf{f}_{bk} = \begin{bmatrix} f_m \\ 0 \\ f_{Tk} \end{bmatrix} \tag{7.33}$$

Where f_m is the tangential component of the aerodynamic force and f_{Tk} is the perpendicular component which defines the thrust in the axial-propeller direction and f_m generates the torques produced by propellers. In order to analyze the blades forces, a differential analysis is made, R blade length is divided in segments dr. Figure 56 shows the segmentation in the spin frame of reference (see 7.1.1).

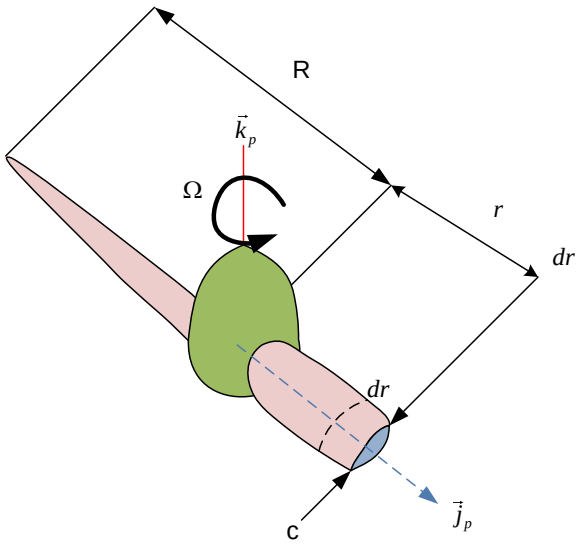

Figure 56. **Blade Segmentation**

Figure 57 illustrates a section of a blade element at r distance in the spin from the propeller spin axis and the forces. The angle between the real blade velocity and the blade cord is called attack angle $\alpha(r)$. The angle between spin plane and the propeller advance

velocity is called $\phi(v_b, r)$ helix angle. The angle between the blade cord and spin plane is called Geometric pitch $\beta(r)$ (Figure 50, Figure 57).

Note

The values of $\phi, \alpha, \beta, \Omega$ must not be confused with Euler angles in other sections.

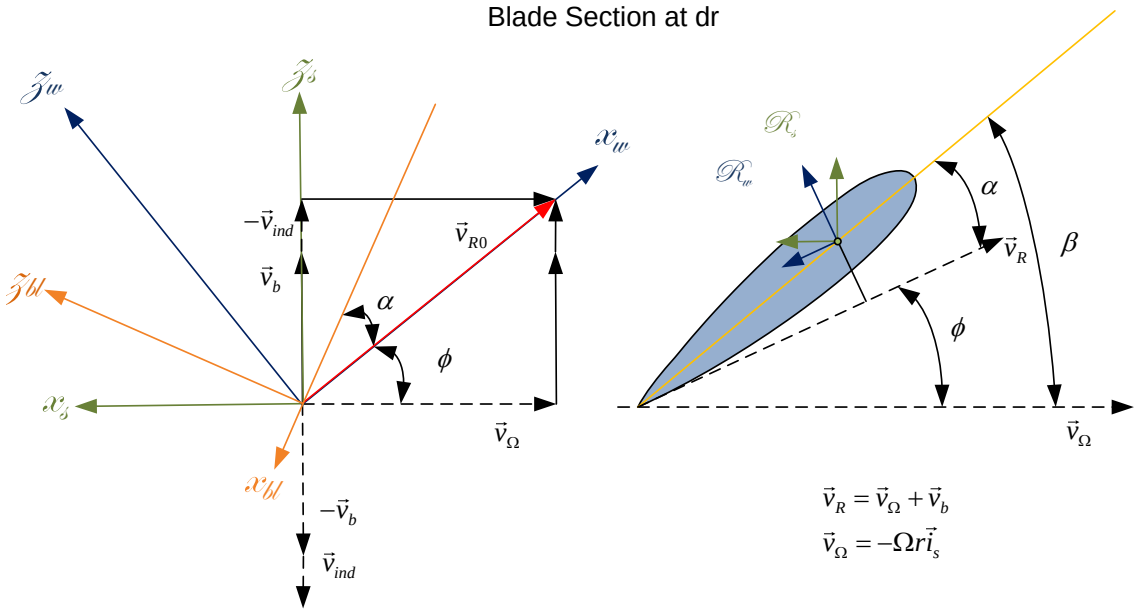

Figure 57. **Geometry of a propeller Blade element**

The lift and drag acting on the transversal section are df_L and df_D, these aerodynamics forces are expressed in thrust and torque components on the transversal section and they are denoted as df_T and dm_T.

7.3.3.1 Thrust analysis

From Aerodynamics point of view is necessary to define $C_L(\phi)$ lift coefficient and $C_D(\phi)$ drag coefficient, these allow to determinate the aerodynamic forces in two components referred to the wind direction. In order to simplify the following analysis the area is expressed as the product of the chord length $c(r)$ and the differential dr [88].

$$d\vec{f}_{akw} = \begin{bmatrix} -df_{Dk} \\ 0 \\ df_{Lk} \end{bmatrix} = \begin{bmatrix} -c_D c(r)\frac{1}{2}\rho v_R{}^2 \\ 0 \\ c_L c(r)\frac{1}{2}\rho v_R{}^2 \end{bmatrix} dr \tag{7.34}$$

The propeller thrust and propeller torque is composed by a lift and a drag component

$$\begin{aligned} &c_\phi = \cos\phi, s_\phi = \sin\phi \\ &d\vec{f}_{aks} = \begin{bmatrix} df_{mk} \\ 0 \\ df_{Tk} \end{bmatrix} = \begin{bmatrix} -c_\phi & 0 & s_\phi \\ 0 & -1 & 0 \\ s_\phi & 0 & c_\phi \end{bmatrix} \begin{bmatrix} -df_D \\ 0 \\ df_L \end{bmatrix} \end{aligned} \tag{7.35}$$

The moment can be obtained in the spin frame of reference.

$$d\vec{f}_{aks} = \begin{bmatrix} df_{mk} \\ 0 \\ df_{Tk} \end{bmatrix} = \begin{bmatrix} -c_\phi & 0 & s_\phi \\ 0 & -1 & 0 \\ s_\phi & 0 & c_\phi \end{bmatrix} \begin{bmatrix} -c_D c\frac{1}{2}\rho v_R{}^2 \\ 0 \\ c_L c\frac{1}{2}\rho v_R{}^2 \end{bmatrix} dr \tag{7.36}$$

The moment is originated by the force components in the plane rotation,

$$d\vec{m}_{Tk} = r\vec{j}_{sj,k} \times df_{mk}\vec{i}_{sj,k} = -rdf_{mk}\vec{k}_{sj,k} \tag{7.37}$$

Therefore,

$$\begin{aligned} df_{Tk} &= \rho v_R{}^2 \left(C_L c_\phi - C_D s_\phi\right) cdr/2 \\ dm_{Tk} &= -\rho v_R{}^2 \left(C_L s_\phi + C_D c_\phi\right) rcdr/2 \end{aligned} \tag{7.38}$$

7.3.4 Combined Blade-Element Theory and Momentum Theory

In general cases the area S is not constant, it changes with the value r. For this case the value of S is assumed constant. Therefore for future analysis the solidity ratio of a propeller σ is define as,

$$\sigma = \frac{n_b cR}{\pi R^2} = \frac{n_b c}{\pi R} \tag{7.39}$$

These equations for force and moment can be solved with numerical analysis (iterations), but in a real time embedded system these kind of implementations are not feasible. To resolve this issue, a previous numerical analysis is made with a set of wind velocities and angular velocities, generating a lookup table, and then to apply the least square method to not only identify the classical parameters but also increase the order of the polynomial and take into account the wind velocity (different to classical approaches). This procedure was performed and analyzed for Borea in the following case of study.

7.3.4.1 Case of Study

The propeller case of study is described in Table 7, these features are based on the Borea quadrotor propellers.

Table 7. Propeller Parameters for simulation

No	Variable	Symbol	Unit	Value	Comments
1	Cord	c	[m]	0.01	0.01
2	Number of blades	n_b	[]	2	2
3	Blade length	R	[m]	0.127	0.127
4	pitch wing	P_a	[m]	0.119	0.177
5	solidity ratio	σ	[]	0.501	0.501
6	Air Density	ρ	[kg/m2]	1	1
7	Lift Slope	CL_0	[1/Rad]		
8	Drag coefficient (Zero Lift)	CD_0	[]	0.02	0.02
9	Oswald efficiency number	e	[]	0.85	0.85
10	Motor velocity	Ω	[rad/s]	630	630

The following simulations show the comparison between two propellers, both of them have the same radius, with different pitch wing,

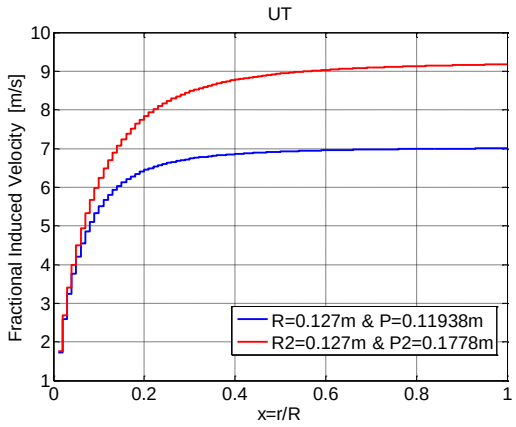

Figure 58. **Fractional Induced Velocity v_{ind} along the blade**

Figure 58 shows the comparison fractional induced velocity, through the blade, Figure 59 shows the pitch angle and inflow angle along the blade with a normalize axis $x = r / R$.

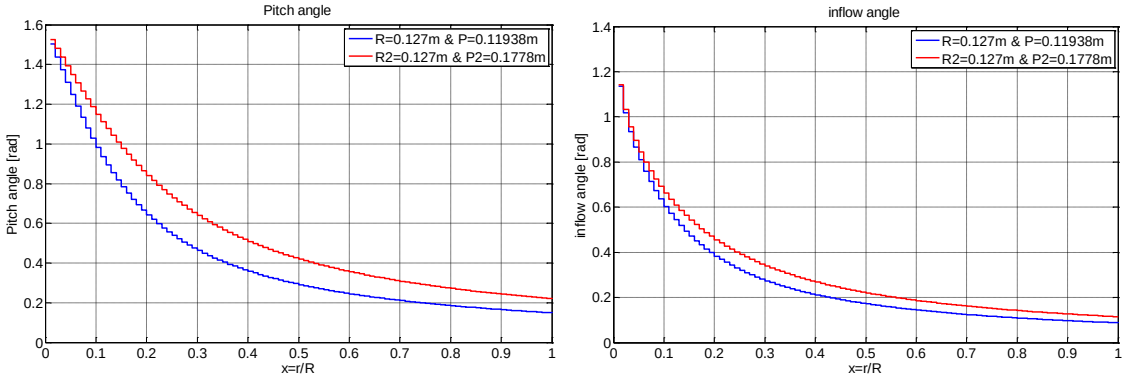

Figure 59. **Pitch angle(left) Inflow angle (Right)**

The force produced by the propeller is determinate by the fractional induced velocity (Figure 60). When the integrations along the blade of the expressions in (7.38) are performed the force and moment obtained in function of the angular velocity and wind velocity.

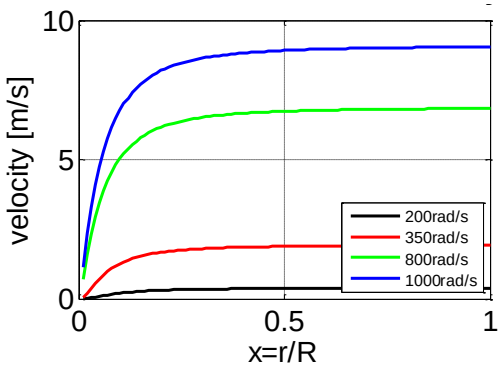

Figure 60. **Fractional Induced Velocity** v_{ind}

The axial force obtained by the integration is shown in Figure 61 and Figure 62 (left).

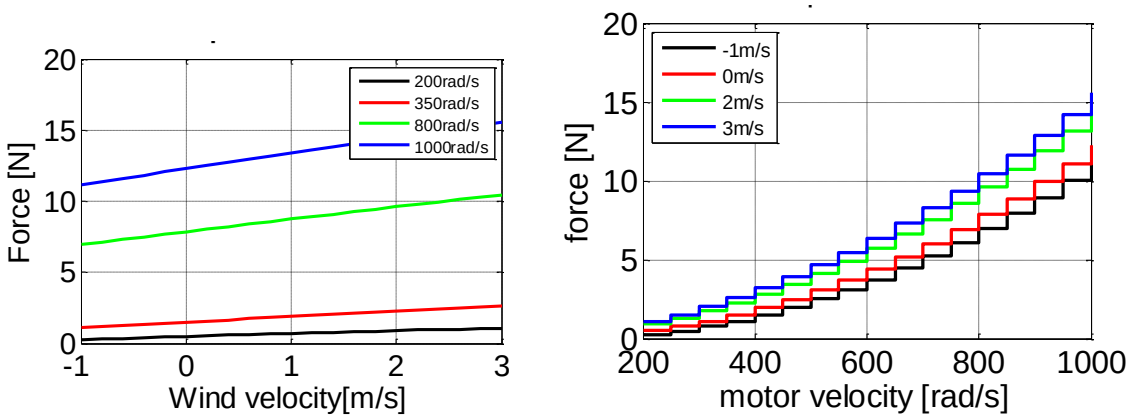

Figure 61. **Thrust vs climb velocity**

The lookup table is generated (Figure 62,left), and with this a least square method is applied to identify a polynomial that summarizes the non linear model in a fast computational implementable model. The resultant model with a maximum error of 5% is shown Figure 62 (right).

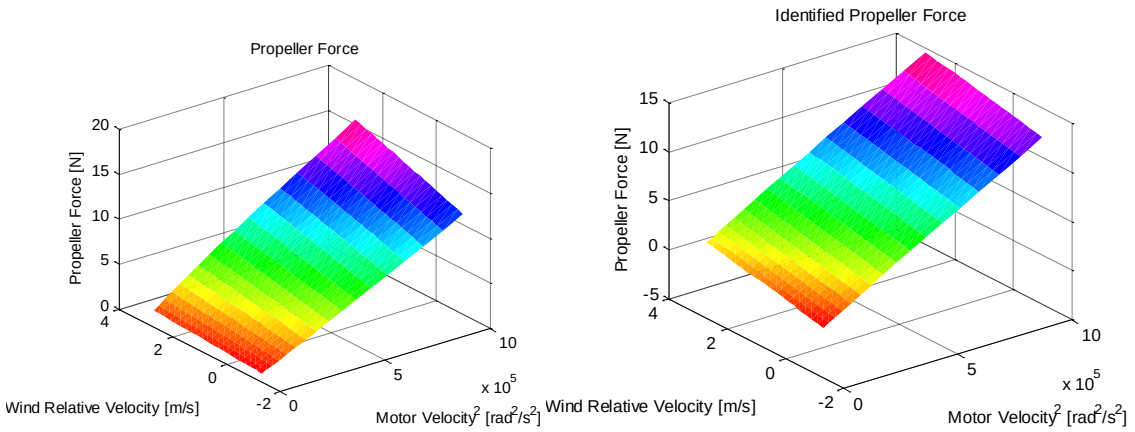

Figure 62. **Theoretical Propeller Force**

7.4 Com Dynamics

The quadrotor analysis has some specific features: The thrust is given by four propeller-motor set. The propeller forces are defined as $f_{pi}, i = 1,2,3,4$; all forces are assumed to be applied in the centre of mass; the quadrotor is an uncompressible body with constant mass.

The Newtown's law equations are applied in the local reference frame, the force is assumed to be applied in the centre of the mass m_q. From the total applied force $\mathbf{f}_q$ that includes the propeller, gravity and disturbances forces, the quadrotor acceleration $\mathbf{a}_q$ is obtained, this is shown in equation (7.40), i.e., the command who defines the velocity $\mathbf{v}_q$ and position $\mathbf{x}_q$

$$\mathbf{f}_q / m_q = \mathbf{a}_q, \mathbf{f}_q = \mathbf{f}_u + \mathbf{f}_g + \mathbf{f}_d \,. \tag{7.40}$$

Therefore quadrotor dynamic model follows the classic mechanics, the main variables in the inertia frame of reference are defined as,

$$\dot{\mathbf{v}}_{ql} = \mathbf{a}_{ql} = \begin{bmatrix} a_{qlx} \\ a_{qly} \\ a_{qlz} \end{bmatrix}, \dot{\mathbf{x}}_{ql} = \mathbf{v}_q = \begin{bmatrix} v_{qlx} \\ v_{qly} \\ v_{qlz} \end{bmatrix}, \mathbf{x}_{ql} = \begin{bmatrix} x_{qlx} \\ y_{qly} \\ z_{qlz} \end{bmatrix}. \tag{7.41}$$

As mentioned before, the quadrotor has a 4 motor-propeller set that generates an input force $\mathbf{f}_{ul}$ in the local frame of reference, it can be defined as,

$$\mathbf{f}_{ul} = \mathbf{f}_{pl1} + \mathbf{f}_{pl2} + \mathbf{f}_{pl3} + \mathbf{f}_{pl4} \,. \tag{7.42}$$

The forces applied to the quadrotor are mainly two, the input force $\mathbf{f}_{ul}$ and the gravity force $\mathbf{f}_{gl}$, since the assumptions are not entirely fulfilled (symmetry and mount discrepancies), disturbance forces are included in the model. The Figure 63 and equation (7.43) summarize the model.

$$\mathbf{f}_q = \mathbf{f}_u + \mathbf{f}_g + \mathbf{f}_d \,. \tag{7.43}$$

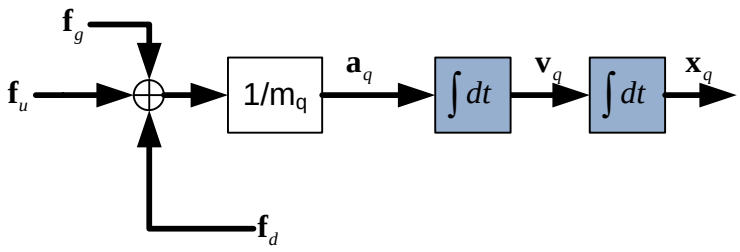

Figure 63. **Com Dynamics**

The equation (7.44) shows that the input force in the body reference frame is applied only on z axis.

$$\mathbf{f}_{ub} = \begin{bmatrix} f_{ubx} \\ f_{uby} \\ f_{ubz} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ f_{uzb} \end{bmatrix} \tag{7.44}$$

When the body reference frame and the inertial reference frame are aligned, the quadrotor can only move along the vertical axis, therefore a tilting is required in order to displace along the horizontal plane. The force command is clearly represented in the body frame of reference, the transformation is mandatory to develop the centre of mass (COM) dynamics.

The gravity in the local reference frame, by definition has only one component on the z axis of the local frame of reference

$$\mathbf{f}_{gl} = \begin{bmatrix} 0 \\ 0 \\ -m_q g \end{bmatrix}, g = 9.81\text{m/s}^2 \tag{7.45}$$

.

The disturbance forces expressed in the local frame of reference are

$$\mathbf{f}_{dl} = \begin{bmatrix} f_{dlx} \\ f_{dly} \\ f_{dlz} \end{bmatrix} \tag{7.46}$$

The command force in the local frame of reference is shown in the equation (7.47). This force has only one component, this allows to reduce the expression of the propeller force components to

$$\begin{bmatrix} f_{ulx} \\ f_{uly} \\ f_{ulz} \end{bmatrix} = \begin{bmatrix} \mathrm{c}_\theta \mathrm{c}_\psi & -\mathrm{c}_\theta \mathrm{s}_\psi & \mathrm{s}_\theta \\ \mathrm{s}_\phi \mathrm{s}_\theta \mathrm{c}_\psi + \mathrm{c}_\phi \mathrm{s}_\psi & -\mathrm{s}_\phi \mathrm{s}_\theta \mathrm{s}_\psi + \mathrm{c}_\phi \mathrm{c}_\psi & -\mathrm{s}_\phi \mathrm{c}_\theta \\ -\mathrm{c}_\phi \mathrm{s}_\theta \mathrm{c}_\psi + \mathrm{s}_\phi \mathrm{s}_\psi & \mathrm{c}_\phi \mathrm{s}_\theta \mathrm{s}_\psi + \mathrm{s}_\phi \mathrm{c}_\psi & \mathrm{c}_\phi \mathrm{c}_\theta \end{bmatrix} \begin{bmatrix} f_{ubx} \\ f_{uby} \\ f_{ubz} \end{bmatrix} \tag{7.47}$$

Finally, the COM dynamics model in the local frame of reference is

$$m_q \begin{bmatrix} \dot{v}_{lx} \\ \dot{v}_{ly} \\ \dot{v}_{lz} \end{bmatrix} = \begin{bmatrix} \mathrm{s}_\theta \\ -\mathrm{s}_\phi \mathrm{c}_\theta \\ \mathrm{c}_\theta \mathrm{c}_\phi \end{bmatrix} f_{ubz} + \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} + \mathbf{f}_{dl} \tag{7.48}$$

.

The equation (7.48) shows the coupling between the attitude and the Com dynamics, in a simple analysis, small tilting in the axis X produces a forces in the axis Y and the same for the opposite case, in the local frame of reference this interconnection can be modified in order to have a coupled model directly from the attitude, the command forces are expressed in the intermediate frame of reference.

$$R_l^{l_i} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{7.49}$$

The command forces pass through the transformation $R_l^{l_i}$ in the local frame of reference; because the axis Z remains equal the gravity is not affected for the transformation.

$$\mathbf{f}_{ul_i} = R_l^{l_i} \cdot R_b^l \cdot \mathbf{f}_{ub} \tag{7.50}$$

$$\mathbf{f}_{ul_i} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathrm{s}_\theta \\ -\mathrm{s}_\phi \mathrm{c}_\theta \\ \mathrm{c}_\theta \mathrm{c}_\phi \end{bmatrix} f_{ubz} \tag{7.51}$$

The command in the local intermediate frame of reference

$$\mathbf{f}_{ul_i} = \begin{bmatrix} f_{ul_i x} \\ f_{ul_i y} \\ f_{ul_i z} \end{bmatrix} = \begin{bmatrix} \mathrm{s}_\phi \mathrm{c}_\theta \\ \mathrm{s}_\theta \\ \mathrm{c}_\theta \mathrm{c}_\phi \end{bmatrix} f_{ubz} \tag{7.52}$$

Finally COM dynamics Model in the local intermediate frame of reference

$$m_q \begin{bmatrix} \dot{v}_{l_i x} \\ \dot{v}_{l_i y} \\ \dot{v}_{l_i z} \end{bmatrix} = \begin{bmatrix} \mathrm{s}_\phi \mathrm{c}_\theta \\ \mathrm{s}_\theta \\ \mathrm{c}_\theta \mathrm{c}_\phi \end{bmatrix} f_{ubz} + \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} + \mathbf{f}_{dl_i} \tag{7.53}$$

The equation (7.53) shows that the vertical model is not modified, therefore the vertical analysis made in the local intermediate frame of references is valid also for the local frame of reference. The horizontal model in the intermediate local frame of reference is used in the following sections.

7.5 Euler Equation of Rotation

The Euler dynamics applied to the quadrotor follows the equation(7.54), where $\boldsymbol{\omega}_q(t)$ is the quadrotor angular velocity, J_q is the quadrotor inertia tensor and $\mathbf{c}_q(t)$ are the torques. This expression takes into account the gyroscopic acceleration in the first term on the right side. For Borea case the inertia tensor J_q and mass m_q were found on [25].

$$\dot{\boldsymbol{\omega}}_b(t) = -J_q^{-1}\boldsymbol{\omega}_b(t) \times J_q\boldsymbol{\omega}_b(t) + J_q^{-1}\mathbf{c}_q(t) \tag{7.54}$$

The block diagram in Figure 64 shows the Euler rotation for Borea project,

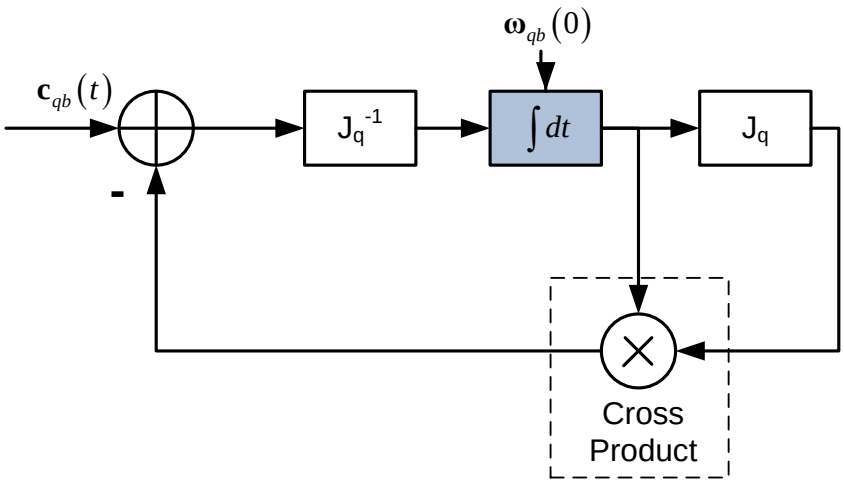

Figure 64. **Euler Dynamics**

The quadrotor torque $\mathbf{c}_q$ is defined by two main effects, the propeller torques, defined by the aerodynamics effects and the propeller forces located in the different points. For discrepancies with the model, caused by the wind or other elements, a disturbance component $\mathbf{c}_d$ is added to the model.

$$\mathbf{c}_q = \mathbf{c}_u + \mathbf{c}_d \tag{7.55}$$

The command torque $\mathbf{c}_u$ is defined by two different sources, the torque produced by the propellers and the torque produced by the propeller forces not applied in the COM, both of them are commanded by the angular rate of the motors. The procedure to obtain the forces and torques in the body frame of reference is called dispatching. Therefore the only missing part to complete the quadrotor model is the update of the attitude representation, i.e., to obtain the angles from the angular velocity, a simple relationship is shown in the equation (7.56).

$$\begin{bmatrix}\omega_{bx}\\ \omega_{by}\\ \omega_{bz}\end{bmatrix} = Z(\psi)Y(\theta)\begin{bmatrix}\dot{\phi}\\ 0\\ 0\end{bmatrix} + Z(\psi)\begin{bmatrix}0\\ \dot{\theta}\\ 0\end{bmatrix} + \begin{bmatrix}0\\ 0\\ \dot{\psi}\end{bmatrix} \tag{7.56}$$

Summarizing, the angular rate can be obtained from the Euler angles $[\phi,\theta,\psi]$ (attitude) and angular velocity ω.

$$\begin{bmatrix}\omega_{bx}\\ \omega_{by}\\ \omega_{bz}\end{bmatrix} = \begin{bmatrix}\mathrm{c}_{\psi_b}\mathrm{c}_{\theta_b} & -\mathrm{s}_{\psi_b} & \mathrm{c}_{\psi_b}\mathrm{s}_{\theta_b}\\ \mathrm{s}_{\psi_b} & \mathrm{c}_{\psi_b} & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}\dot{\phi}_b\\ \dot{\theta}_b\\ \dot{\psi}_b\end{bmatrix} \tag{7.57}$$

In order to represent the rotations and attitude kinematics the quaternion-rotation representations are used. The quaternions are the generalized complex numbers [91], these elements are other way of the attitude representation.

$$\boldsymbol{q} = \begin{bmatrix}q_0\\ \mathbf{q}\end{bmatrix} = \begin{bmatrix}\cos(\varphi/2)\\ \mathbf{e}\sin(\varphi/2)\end{bmatrix} = \begin{bmatrix}q_0\\ q_1\\ q_2\\ q_3\end{bmatrix} \tag{7.58}$$

A quaternion can represent the attitude through the Euler parameters $(\mathbf{e},\varphi)$, where $-\pi \le \varphi < \pi$, and $\mathbf{e}$ is the Rodriguez vector [92] i.e. $\mathbf{e}$ is the unique rotation axis in which the vector is rotate a φ angle, therefore the quaternion should have unitary norm to represent transformations and rotations. The conversion from quaternion to matrix rotation is obtained from the Rodriguez formula,

$$\begin{aligned}\boldsymbol{q} &= \begin{bmatrix}\cos\phi/2\\ \mathbf{e}\sin\phi/2\end{bmatrix} = \begin{bmatrix}q_0\\ \mathbf{q}\end{bmatrix}\\ R(\boldsymbol{q}) &= \left(q_0^2 - \mathbf{q}^T\mathbf{q}\right)I + 2\mathbf{q}\mathbf{q}^T + 2q_0\mathbf{q}\times\end{aligned}, \tag{7.59}$$

where $\mathbf{q}\times$ is the skew matrix of the unitary vector $\mathbf{q}$

$$\mathbf{q}\times = \begin{bmatrix}0 & -q_3 & q_2\\ q_3 & 0 & -q_1\\ -q_2 & q_1 & 0\end{bmatrix}. \tag{7.60}$$

The quaternion has a special advantage when is used in kinematic differential equations, it is possible to know the derivative of the quaternion from the current quaternion and the

angular velocity vector, this allows considering this quaternion as a non linear state [92]. The equation is shown below, where the symbol $\otimes$ represents the quaternion product.

$$\dot{\boldsymbol{q}}(t)=\frac{1}{2}\boldsymbol{q}(t)\otimes\begin{bmatrix}0\\ \omega(t)\mathbf{e}(t)\end{bmatrix}=\frac{1}{2}\boldsymbol{q}(t)\otimes\begin{bmatrix}0\\ \boldsymbol{\omega}(t)\end{bmatrix}=\frac{1}{2}\boldsymbol{q}(t)\otimes\boldsymbol{\omega}(t),\ \omega(t)=\left|\boldsymbol{\omega}(t)\right|, \quad (7.61)$$

The angular velocity is obtained from the Euler dynamics, therefore this set of systems allows the simulator to update the quaternion and hence the rotation matrix.

7.6 Dispatching

The forces of each propeller denoted as $T_i \quad i=1,2,3,4$ are represented in the input propeller force vector f_{up} as,

$$\mathbf{f}_{up}=\begin{bmatrix}f_{pb1}\\ f_{pb2}\\ f_{pb3}\\ f_{pb4}\end{bmatrix}. \quad (7.62)$$

There is a transformation from the propeller force values to the body reference frame, this is represented with the matrix V. The following equation shows the general way to obtain the force body values.

$$\mathbf{f}_{bu}=\begin{bmatrix}f_{bux}\\ f_{buy}\\ f_{buz}\end{bmatrix}=V\mathbf{f}_{up} \quad (7.63)$$

For the quadrotor case the propellers also can give axial thrush therefore the totality of force is applied in the z axes. The matrix V which represents this case is

$$V = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}. \tag{7.64}$$

Each propeller gives a force and torque, the torques have two sources. One is obtained from the propeller dynamics, and the other one due to the forces not applied in the center of mass. To identify this torque the position of each propeller is required, the matrix V_L contains the location of each propeller in the body frame of reference just as shown next, where d is the distance thought the arms between the propeller centers. And h is the distance along the vertical axis from the geometric center and propeller center.

$$V_L = \begin{bmatrix} d/2 & 0 & -d/2 & 0 \\ 0 & -d/2 & 0 & d/2 \\ -h & -h & -h & -h \end{bmatrix}, \tag{7.65}$$

The torque command is shown in the following equation,

$$\mathbf{c}_u = \begin{bmatrix} m_x \\ m_y \\ m_z \end{bmatrix} \tag{7.66}$$

It is not possible to apply horizontal forces; therefore, the only controllable commands are the axial force $\mathbf{f}_{bz}$ and the torques. The controllable command vector $\mathbf{P}_{zm}$ is defined as,

$$\mathbf{P}_{zm} = \begin{bmatrix} f_z \\ m_x \\ m_y \\ m_z \end{bmatrix} = B_{zm}\mathbf{f}_{bu}. \tag{7.67}$$

Classical approaches ignore the relative wind velocity and the remaining component proportional to the square of the angular velocity, therefore the relationship between the

moment and the force can be expressed as $k_d = \mathrm{m}_{pbi} / f_{pbi}$, then the matrix B_{zm} can be written as

$$B_{zm} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & d/2 & 0 & -d/2 \\ -d/2 & 0 & d/2 & 0 \\ -k_d & k_d & -k_d & k_d \end{bmatrix}. \tag{7.68}$$

The dispatching allows the operator to traduce the requirements of axial force and moments to propeller angular velocities, approximating the force to a single element proportional to this rate the other components will be added to the disturbances. B_{zm} is a non singular matrix which allows to determine the propeller forces from the requirements.

$$\mathbf{f}_{bu} = B_{zm}{}^{-1}\mathbf{P}_{zm}. \tag{7.69}$$

The following matrix will be used to transform the requirements to propeller forces and then the angular velocities.

$$B_{zm}{}^{-1} = \begin{bmatrix} \frac{1}{4} & 0 & -\frac{1}{d} & -\frac{1}{4k_d} \\ \frac{1}{4} & \frac{1}{d} & 0 & \frac{1}{4k_d} \\ \frac{1}{4} & 0 & \frac{1}{d} & -\frac{1}{4k_d} \\ \frac{1}{4} & -\frac{1}{d} & 0 & \frac{1}{4k_d} \end{bmatrix}. \tag{7.70}$$

7.6.1 Simulator results

The following test shows typical requirements from the control unit. There are moment requirements in Figure 65 and force requirements in Figure 66, the force applied is divided into steps, and the moment requirements are polynomial type, the origin of the requirements is explained in the guidance chapter, and the real values of the requirements

are found in the control law chapter. The main point in this section is to define the propellers commands at each discrete time from the force and torques requirements.

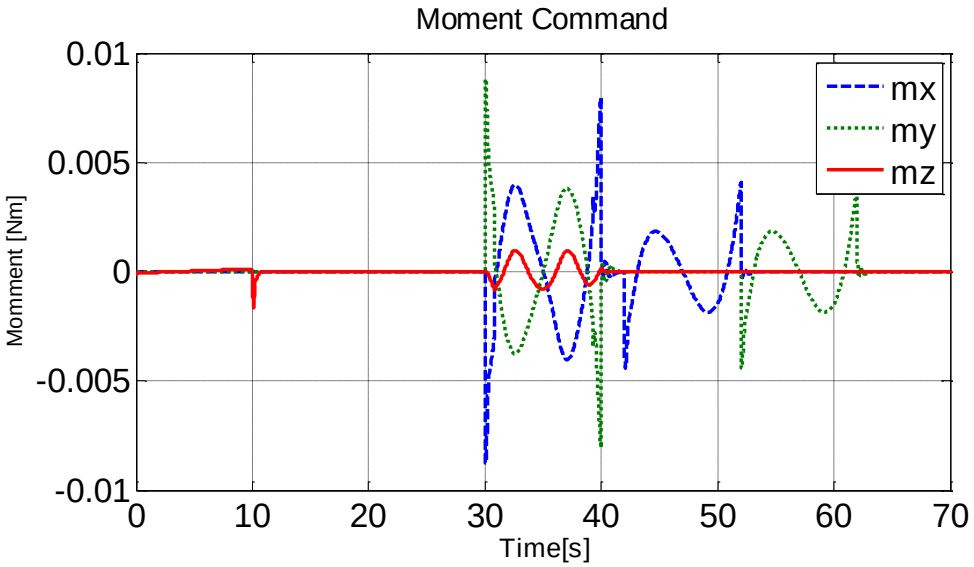

Figure 65. **Moment Requirements**

The force requirements always guarantee at least the hover condition, in order to accomplish this task the gravity is compensated, then when the quadrotor is not inclined and is in hover condition the force will be equal in magnitude and opposite to the gravity.

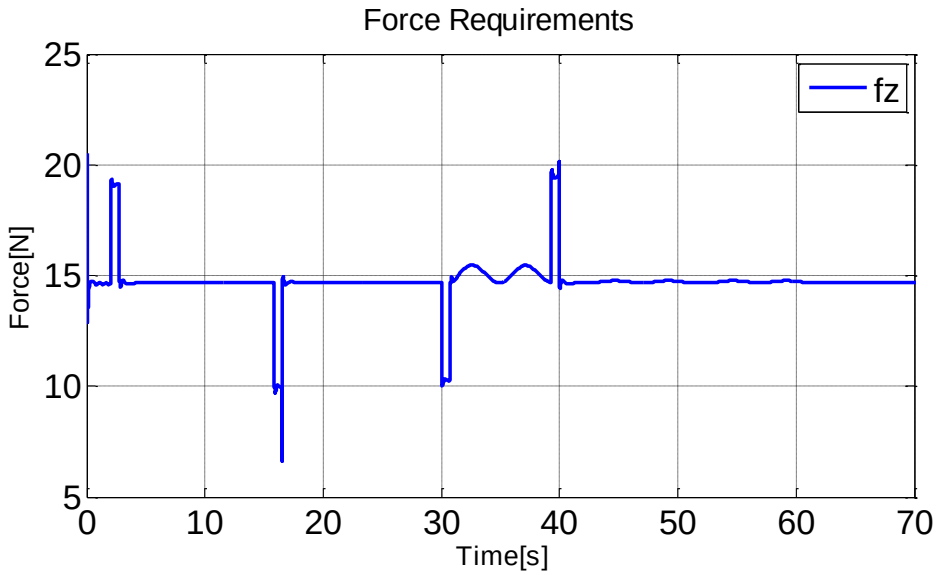

Figure 66. **Force Requirements**

The Figure 67 and Figure 68 show the former requirements expressed as propeller force requirements, the instant average value of the propellers is equal to the force requirements, on another hand the differences of the force represent the moments.

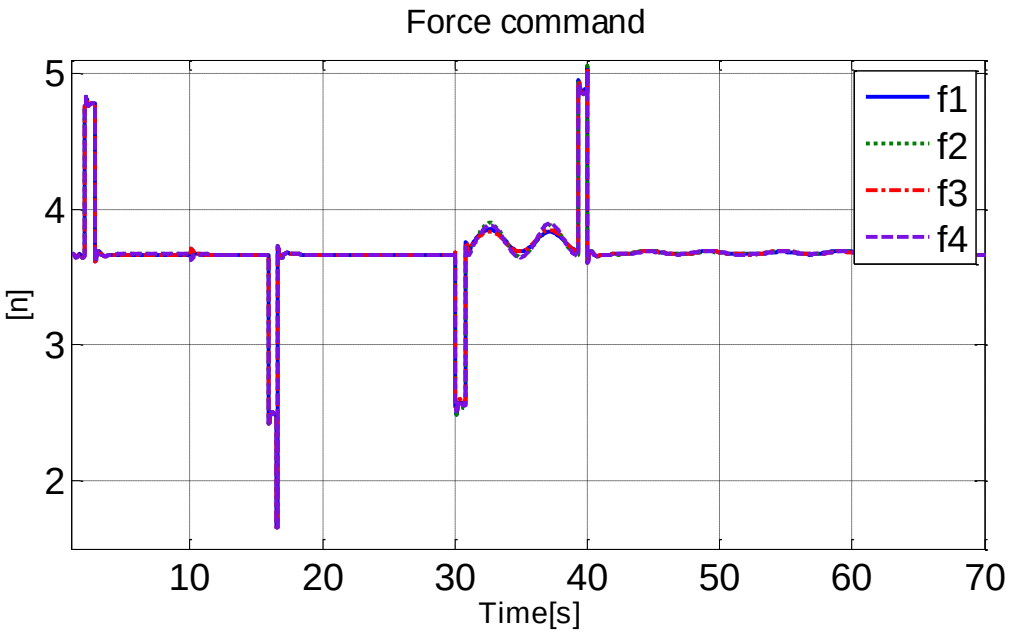

Figure 67. **Propeller Force**

From propeller point of view, the moment requirements are not as demanding as the force, a change in the moment only produces small variations in the quadrotor propeller, the Figure 68 shows an amplified version of the propeller forces, the diference between the propellers forces are proportional to required moments.

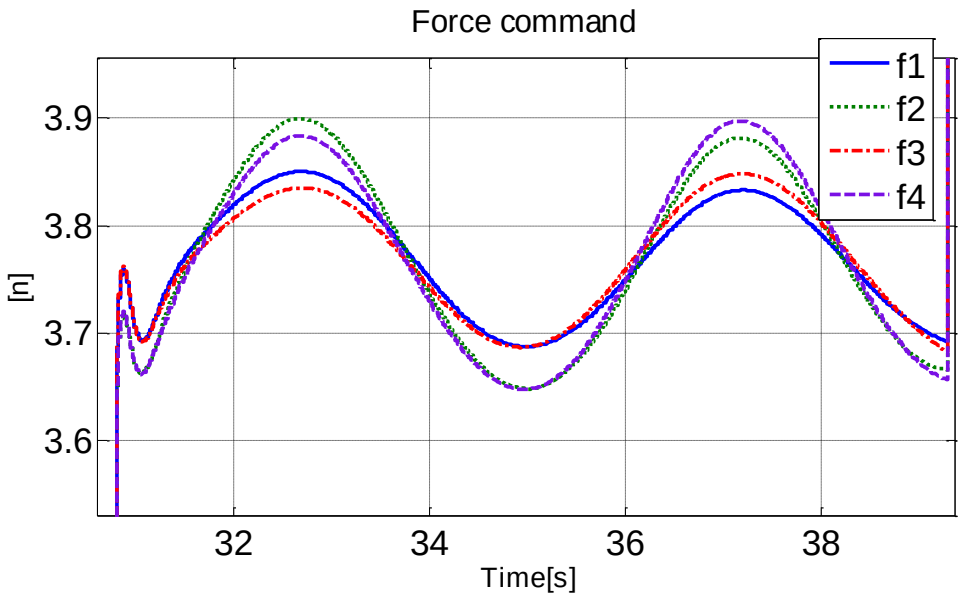

Figure 68. **Propeller force (Amplified)**

The full simulator model is shown in Figure 69, this includes the displacement in the local reference frame, the Euler dynamics and quaternion kinematics.

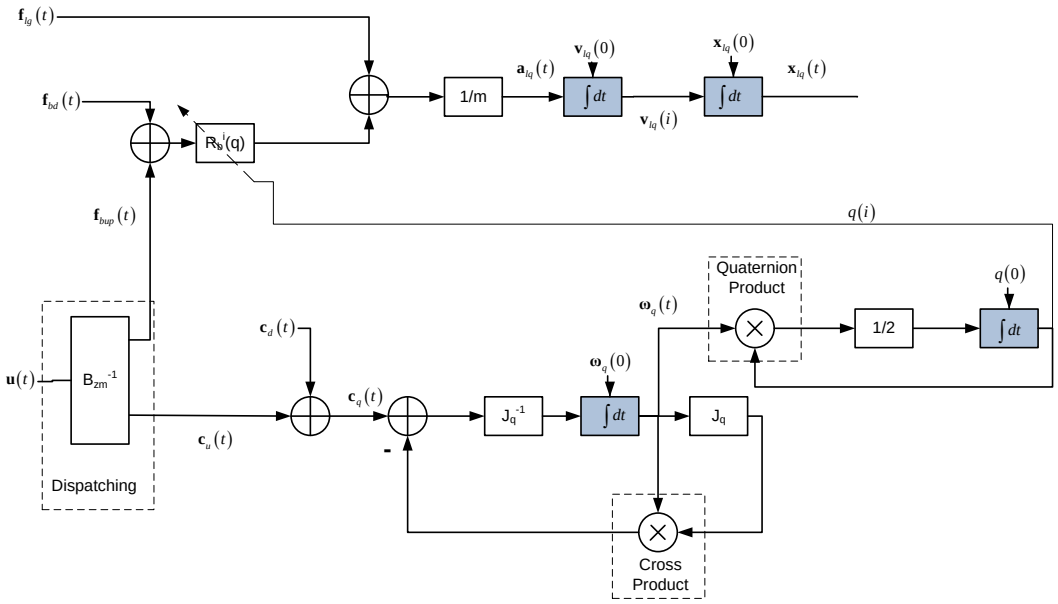

Figure 69. **Quadrotor Dynamic and Kinematic Diagram**

8. EMBEDDED MODEL FOR BOREA QUADROTOR

Since all the state variables are not available (for instance the disturbance terms to be rejected) state estimation is mandatory and facilitated by the embedded model. Thinking to outdoor flight: IMU, GPS, ultrasound sensor and magnetometer [75] are typically used. For indoor flight GPS is not used. The navigation is organized in to different state predictors, consisting of the embedded model that is fed back by noise estimator, driven by the model error (measurement minus model output). The noise estimators are analyzed in a discrete time. all state units are meters.

The model error is the sole accessible measure of the uncertainty. Its current value summarizes the past discrepancies that have not been saved in the embedded model. The model error can be elaborated and accumulated in disturbance states. The residual discrepancies are used to reduce the model error which is must brought to be bounded (internal stability).

This research presents an approximation to the quadrotor problem, through a feedback linearization on the horizontal model, the horizontal displacement model is connected with the attitude model through a feedback linearization, the non lineal problem is transformed into a linear problem disregarding the angle value, and allows the problem to be regarded as a unique problem instead of to divide into two control levels.

The embedded models are commanded by two input vectors: the known command computed at each step k by the control unit, and the noise, that by definition, is unknown and unpredictable.

8.1 Timing considerations

The analog model is summarized below, the command f_{ulz} is limited in frequency by the electronic speed control (ESC) since it uses standard servo signals, i.e., the command is applied at 50Hz ($T_{con} = 20ms$), therefore all embedded models are discrete, this fact introduces a quantization error and hence disturbances in the controllable model.

The Model is discretized with $\dot{\mathbf{x}}(t)=\left(\mathbf{x}(t)-\mathbf{x}(t-T)\right)/T$, in order to simplify the diagrams, digital integrators are used.

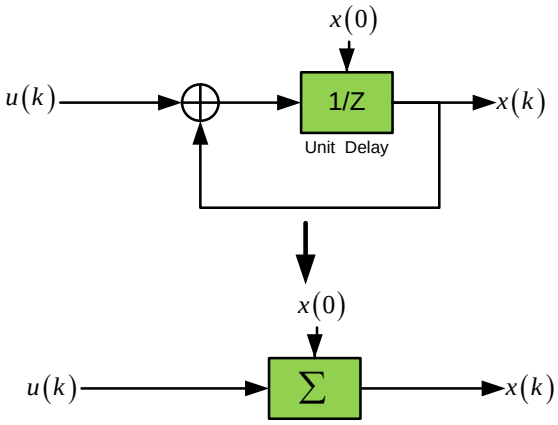

Figure 70. **Digital integrator Scheme**

8.2 Vertical Embedded Model

The next equation represents the vertical analog model,

$$m_q a_{zl} = \mathrm{c}_\phi \mathrm{c}_\theta \, f_{ubz} - g + f_{dlz} \,. \tag{8.1}$$

The embedded model only considers the rigid motion, confining flexible dynamics to the error model, the total disturbance acceleration is modelled as a first order dynamics.

The disturbances only affect the acceleration, they cannot have disturbances in the velocity. Vertical EM equations are summarized as

$$\begin{bmatrix} \hat{x}_z \\ \hat{v}_z \\ D_1 \end{bmatrix}(k+1) = \left[\begin{array}{cc:c} 1 & 1 & 0 \\ 0 & 1 & 1 \\ \hdashline 0 & 0 & 1 \end{array}\right] \begin{bmatrix} \hat{x}_z \\ \hat{v}_z \\ D_1 \end{bmatrix}(k) + \begin{bmatrix} 0 \\ 1 \\ \hdashline 0 \end{bmatrix} u(k) + \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ \hdashline 1 & 0 \end{bmatrix} \begin{bmatrix} w_{1z} \\ w_{2z} \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ \hdashline 0 \end{bmatrix} g \,. \tag{8.2}$$

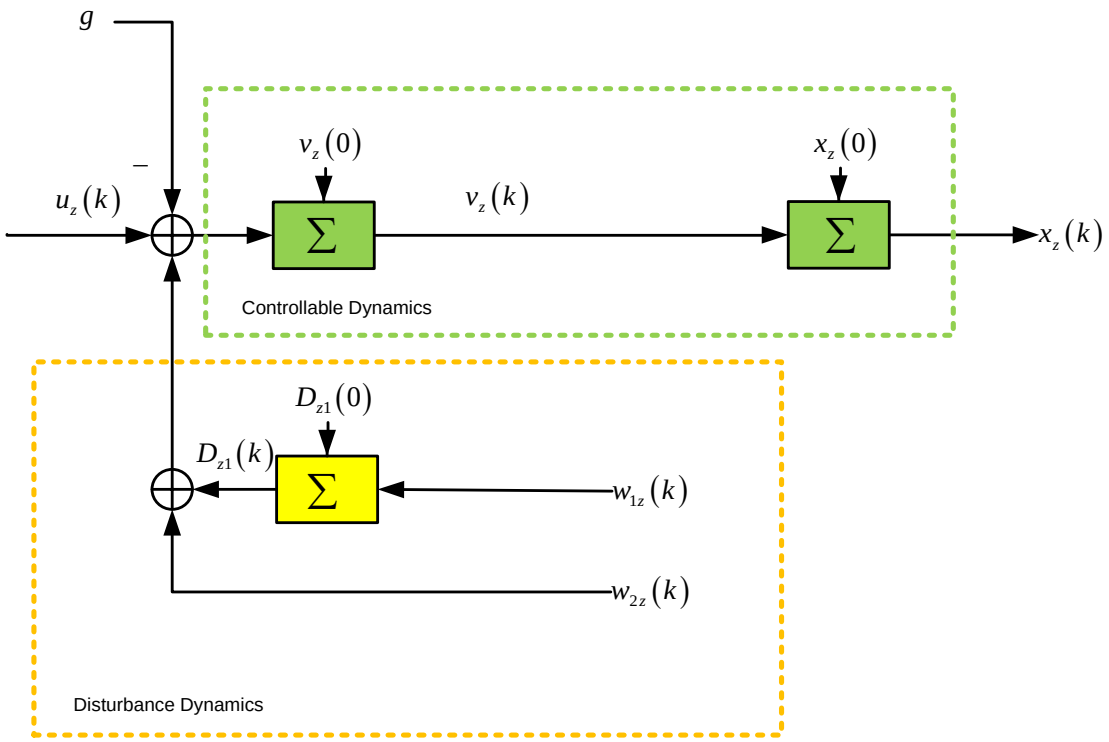

Figure 71. **Diagram block of Vertical EM**

8.3 Horizontal Embedded Model

8.3.1 XY Model

The vertical analog model is given by

$$m_q \begin{bmatrix} a_{xl} \\ a_{yl} \end{bmatrix} = \begin{bmatrix} \mathrm{s}_\phi \mathrm{c}_\theta \\ \mathrm{s}_\theta \end{bmatrix} f_{ubz} + \begin{bmatrix} f_{dxl} \\ f_{dyl} \end{bmatrix}. \tag{8.3}$$

The model reveals a connection between the Z model and XY model, both of them are defined by the axial command and the attitude, if the attitude is known then the command f_{ubz} is defined to accomplish the vertical requirements and consequently for the XY model this command is previously defined. The actual command for the horizontal component is the attitude (ϕ,θ), a new state $\mathbf{q}$ is defined as

$$\mathbf{q}_l = \begin{bmatrix} q_{lx} \\ q_{ly} \end{bmatrix} = \begin{bmatrix} s_\varphi c_\theta \\ s_\theta \end{bmatrix}. \tag{8.4}$$

The attitude is modified through the 3D torques $\mathbf{c}_u$ also defined by the propellers. The new state can be controlled by torques, hence by state transformation a horizontal linear model is obtained that includes the attitude and horizontal position. In the intermediate local frame of reference a feedback linearization is done [12], [22].

$$\mathbf{\Omega}_l = \dot{\mathbf{q}}_l \tag{8.5}$$

$$\mathbf{\Omega}_l = \begin{bmatrix} \Omega_{lx} \\ \Omega_{ly} \end{bmatrix} = \begin{bmatrix} \cos\varphi & -\sin\varphi\sin\theta \\ 0 & \cos\theta \end{bmatrix} \begin{bmatrix} \omega_{sx} \\ \omega_{sy} \end{bmatrix} \tag{8.6}$$

Where $\omega_{sx}, \omega_{sy}, \omega_{sz}$ are defined as,

$$\begin{bmatrix} \omega_{sx} \\ \omega_{sy} \\ \omega_{sz} \end{bmatrix} = \begin{bmatrix} c_\theta \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} - s_\theta \dot{\varphi} \end{bmatrix}, \mathbf{\omega}_s = \begin{bmatrix} \omega_{sx} \\ \omega_{sy} \\ \omega_{sz} \end{bmatrix} = Z(\psi) \begin{bmatrix} \omega_{bx} \\ \omega_{by} \\ \omega_{bz} \end{bmatrix} \tag{8.7}$$

The local frame of reference is a state transformation, and then a command transformation is applied, the following equation shows the relationship between the new linear model with the quadrotor attitude and angular velocity,

$$\mathbf{\Omega}_i = \begin{bmatrix} \Omega_{ix} \\ \Omega_{iy} \end{bmatrix} = \begin{bmatrix} \cos\varphi & -\sin\varphi\sin\theta \\ 0 & \cos\theta \end{bmatrix} Z(\psi) \begin{bmatrix} \omega_{bx} \\ \omega_{by} \end{bmatrix}. \tag{8.8}$$

Also the command transformation requires the attitude, the following equations show the transformation between the quadrotor command and the new model command.

$$\begin{bmatrix} f_{uix} \\ f_{uiy} \end{bmatrix} = \begin{bmatrix} \frac{c_\psi}{c_\varphi} & \frac{s_\psi}{c_\theta} + c_\psi t_\varphi t_\theta \\ -\frac{s_\psi}{c_\varphi} & \frac{c_\psi}{c_\theta} - s_\psi t_\varphi t_\theta \end{bmatrix} \mathbf{f}_{ul} \tag{8.9}$$

In the horizontal dynamics the disturbances are placed in the acceleration and in Ω through the torques. By following the same procedure of the vertical dynamics the disturbances are set in these points, the order of the disturbance dynamics is set to three.

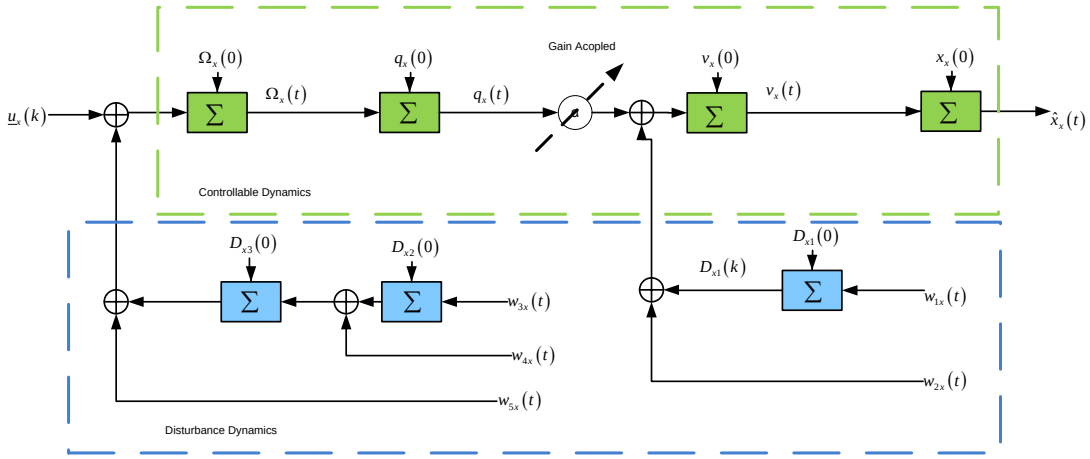

Figure 72. **Diagram Block of the Horizontal EM**

The equation bellow summarizes the EM of the horizontal model, the variable alpha α is determined by the vertical command.

$$\begin{bmatrix} q_x \\ \Omega_x \\ x_x \\ v_x \\ D_1 \\ D_2 \\ D_3 \end{bmatrix}(k+1) = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ \alpha & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} q_x \\ \Omega_x \\ x_x \\ v_x \\ D_1 \\ D_2 \\ D_3 \end{bmatrix}(k) + \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} u_x(k) + \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} w_{x1} \\ w_{x2} \\ w_{x3} \\ w_{x4} \\ w_{x5} \end{bmatrix}, \tag{8.10}$$

where

$$\alpha = f_{lbz} = f_{ubz} / \mathrm{c}_\phi \mathrm{c}_\theta . \tag{8.11}$$

The command and measures transformations are made with the attitude prediction at each time step. The prediction is available thanks to the noise estimator explained in the next chapter.

8.4 Spin Model

The former analysis links the attitude with the displacement through two Euler angles, third Euler angle defines the spin of the quadrotor which can be controlled since the quadrotor allows controlling the three torques. There are not objectives related with the spin, so this variable is controlled only to bring it to zero value. The spin embedded model is

$$\begin{bmatrix} \psi \\ \delta\psi \\ D_{\phi 1} \end{bmatrix}(k+1) = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \psi \\ \omega_{\psi} \\ D_{\phi 1} \end{bmatrix}(k) + \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} u(k) + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} w_{1\phi} \\ w_{2\phi} \end{bmatrix}. \tag{8.12}$$

The variable $\delta\psi$ is not directly the angular velocity ω_{bz}.

9. REFERENCE GENERATOR FOR BOREA QUADROTOR

9.1 Vertical Reference Generator

The vertical EM is linear discrete limited only by linear constrains, (maximum acceleration, maximum velocity), in order to develop the reference generator only the controllable dynamics is used, therefore for the vertical guidance is selected as a bang-bang strategy [93]. This solution consists in the application of the maximum acceleration up to top velocity and then holds.

Remark:

The guidance defines only the reference command, from the control point of view is the open loop command to arrive to desired position. in this chapter a control law is not implemented.

9.2 Horizontal Reference Generator

To develop the lateral reference generator only the controllable dynamics is used, in order to avoid abrupt changes in the attitude model a polynomial strategy is used in the vertical reference generator.

The following analysis is made for single axes; both axes of the horizontal movement track the same reference generator with an independent target. Assuming the α value known the model can be consider as a chain of digital integrators, to simplify analysis the reference generator is developed in continues time and then discretized. After the procedure, the states Ω, q consider the α value. The states follow polynomial manoeuvre, the command $a_x(t)$ is.

$$a(t) = \begin{bmatrix} 1 & t & t^2 & t^3 \end{bmatrix} \mathbf{a} \,. \tag{9.1}$$

The states track a trajectory as is show in the following equation,

$$\begin{bmatrix}\Omega\\ q\\ v\\ x\end{bmatrix}(t)=\begin{bmatrix}\Omega(0)\\ q(0)+\Omega(0)t\\ v(0)+q(0)t+\Omega(0)t^2/2\\ x(0)+v(0)t+q(0)t^2/2+\Omega(0)t^3/6\end{bmatrix}+ \\ +\begin{bmatrix}t & t^2/2 & t^3/3 & t^4/4\\ t^2/2 & t^3/6 & t^4/12 & t^5/20\\ t^3/6 & t^4/24 & t^5/60 & t^6/120\\ t^4/24 & t^5/120 & t^6/360 & t^7/840\end{bmatrix}\begin{bmatrix}a_0\\ a_1\\ a_2\\ a_3\end{bmatrix} \tag{9.2}$$

The main scope is to take the horizontal position from any origin point to a desired position, in order to develop an adaptive strategy, the initial conditions define the vector $\mathbf{x}_0(t)$, i.e., the strategy takes the current measured states to generate a new trajectory, the elements of the reference generator are time functions.

$$\begin{bmatrix}\Omega\\ q\\ v\\ x\end{bmatrix}(t)=\mathbf{x}_0(t)+ \\ +\begin{bmatrix}t & 0 & 0 & 0\\ 0 & t^2/2 & 0 & 0\\ 0 & 0 & t^3/6 & 0\\ 0 & 0 & 0 & t^4/24\end{bmatrix}\begin{bmatrix}1 & 1/2 & 1/3 & 1/4\\ 1 & 1/3 & 1/6 & 1/10\\ 1 & 1/4 & 1/10 & 1/20\\ 1 & 1/5 & 1/15 & 1/35\end{bmatrix}\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & t & 0 & 0\\ 0 & 0 & t^2 & 0\\ 0 & 0 & 0 & t^3\end{bmatrix}\begin{bmatrix}a_0\\ a_1\\ a_2\\ a_3\end{bmatrix} \tag{9.3}$$

The following expression shows the general state responses to a polynomial command

$$\begin{bmatrix}\omega(t)\\ q(t)\\ v(t)\\ x(t)\end{bmatrix}=\mathbf{x}_0(t)+T_1AT_2\mathbf{a}, a(t)=\begin{bmatrix}1 & t & t^2 & t^3\end{bmatrix}\mathbf{a} \tag{9.4}$$

.

The objective of the manoeuvre is to displace the quadrotor. As it was explained before, the initial conditions vector is assumed or known from the sensors, it is clear that in stable cases, the states Ω, q, v are equal to zero.

$$T_2^{-1}A^{-1}T_1^{-1}\left(\mathbf{x}(t)-\mathbf{x}_0(t)\right)=\mathbf{a}. \tag{9.5}$$

The general time depending solution is

$$T_1^{-1}=\begin{bmatrix} 1/t_f & 0 & 0 & 0 \\ 0 & 2/{t_f}^2 & 0 & 0 \\ 0 & 0 & 6/{t_f}^3 & 0 \\ 0 & 0 & 0 & 24/{t_f}^4 \end{bmatrix}, T_2^{-1}=\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1/t_f & 0 & 0 \\ 0 & 0 & 1/{t_f}^2 & 0 \\ 0 & 0 & 0 & 1/{t_f}^3 \end{bmatrix}. \quad (9.6)$$

If the manoeuvre time is predefined, then it is possible to get the coefficients **a** , the algorithm allows the operator to define the value of all states. The results are also valid in discrete time, it is only required to sample the state vector (9.4).

In order to guarantee the position is constant, the reference generator is also implemented with a state machine, while the quadrotor is in the desirable states the output of the reference generator output remains constant. If the operator changes the references, the polynomial command is applied, since the manoeuvre is time-fixed and the operator can change the references at any moment. An update of the reference is made and the process is restarted.

Remark

The guidance is made in continuous time and then sampled at Tcon (20ms), the differences between the EM states and references states will be estimated in the navigation.

9.2.1 Simulations results

A particular case of the guidance is displacement in only one axis Δx with angular rate, attitude and velocity equal to zero. This means that the quadrotor starts and finishes in stable positions,

$$T_2^{-1}A^{-1}T_1^{-1}\left(\mathbf{x}(t)-\mathbf{x}_0(t)\right)=\mathbf{a}$$
$$\left(\mathbf{x}(t)-\mathbf{x}_0(t)\right)=\begin{bmatrix} 0 \\ 0 \\ 0 \\ \Delta x \end{bmatrix} . \quad (9.7)$$

The coefficients are obtained with a final time t_f equal to 10 seconds. An example is shown bellow.

$$\mathbf{a} = \begin{bmatrix} \alpha_1 \\ \alpha_2 / t_f \\ \alpha_3 / {t_f}^2 \\ \alpha_4 / {t_f}^3 \end{bmatrix} 24\Delta x / t^4 = \begin{bmatrix} \alpha_1 = 35 \\ \alpha_2 = -420 \\ \alpha_3 = 1050 \\ \alpha_4 = -700 \end{bmatrix}. \tag{9.8}$$

And finally the command reference is complete,

$$\underline{a}_x(t) = \begin{bmatrix} 1 & t & t^2 & t^3 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 / T \\ \alpha_3 / T^2 \\ \alpha_4 / T^3 \end{bmatrix} 24\Delta x / T^4 = 24\Delta x \left(\alpha_1 + \alpha_2 t / T + \alpha_3 (t/T)^2 + \alpha_4 (t/T)^3 \right) / T^4 \tag{9.9}$$

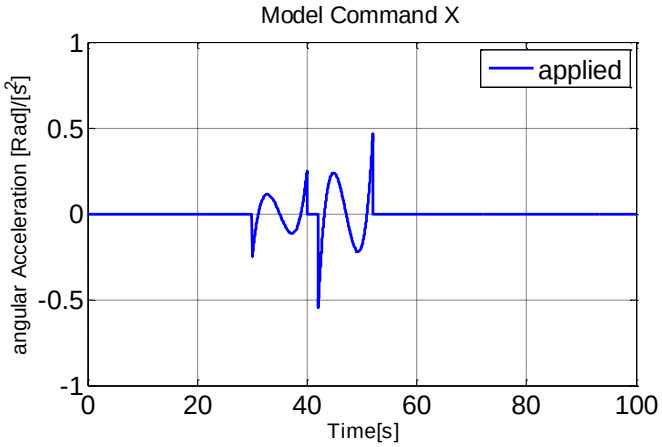

Figure 73. **Reference Command X**

Figure 74 shows an example of the algorithm where the operator demands a displacement of 30 m then a displacement of 15 m, the command obtained with the reference generator is shown in Figure 73.

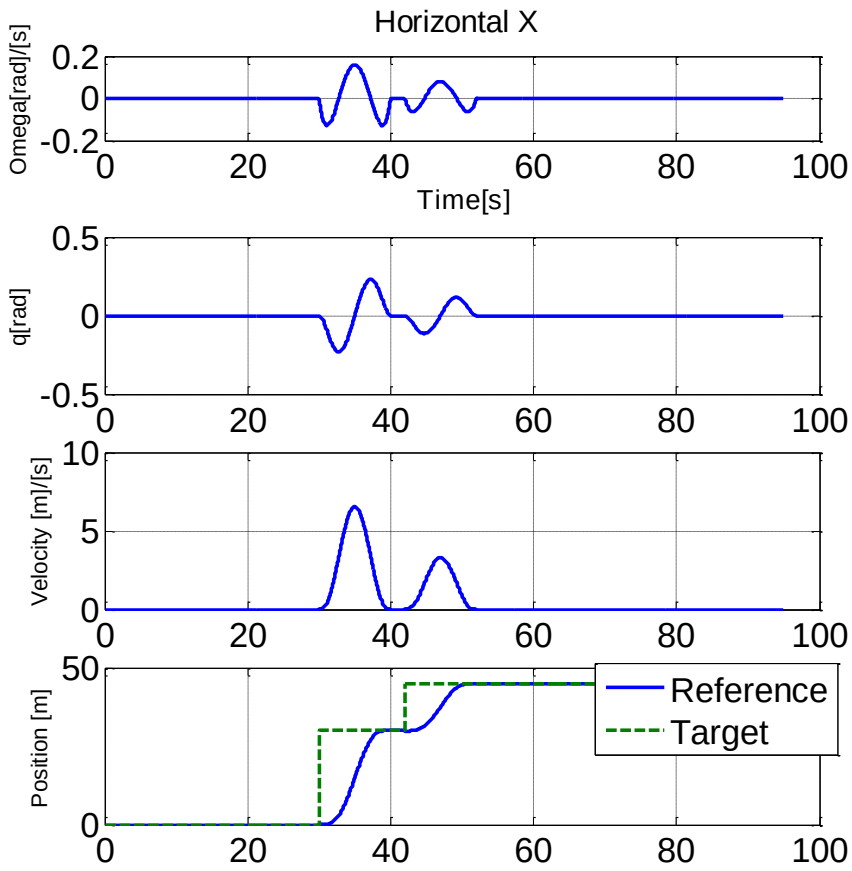

Figure 74. **Horizontal Reference Generator**

Remark: the states Ω_x, q_x consider the α value, this value can be estimated from the vertical command and attitude. The following chapter explains the prediction of these values.

At the beginning of each manoeuvre the current states cannot be the same than the reference states, therefore an adaptive guidance is made, this allows to the reference to be update at the beginning the manoeuvres with the estimated states from the navigation, the same strategy was implemented in the horizontal guidance. The Figure 75 shows a example where the polynomial reference value is updated to the simulated measures, this adaptive feature allows the control to start in not ideal conditions, if the disturbances modify the states, the reference generators are able to get the current states.

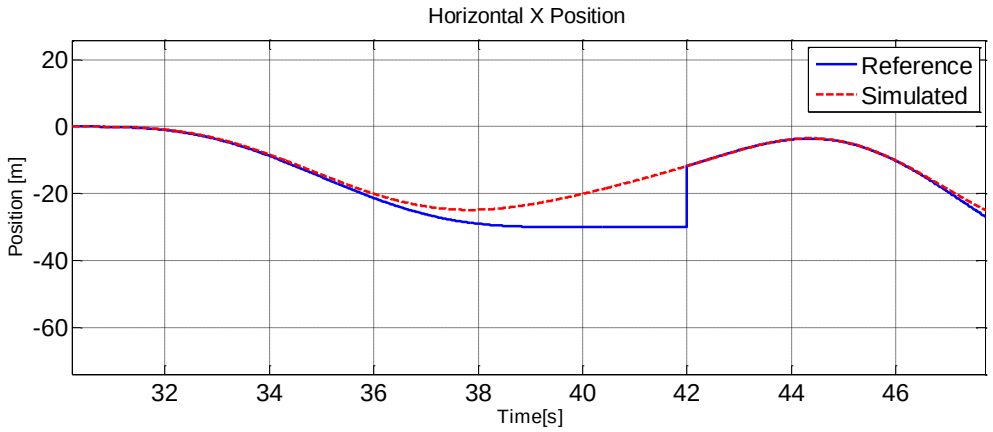

Figure 75. **Update horizontal reference**

Remark: the profiles shown in Figure 74 are defined by the manoeuvre time, whether the value of this variable changes the amplitudes are modified. In order to avoid command saturations, an appropriate time manoeuvre is selected, this allows relaxing the force requirements.

9.3 Spin Reference Generator

This reference generator is also made with the polynomial scheme. The control torque required to this reference generator is of second order, i.e., a simple ramp signal,

$$\dot{\omega}_{\psi}(t) = \begin{bmatrix} 1 & t \end{bmatrix} \begin{bmatrix} a_{s0} \\ a_{s1} \end{bmatrix}. \tag{9.10}$$

By following the same procedure for the previous reference generator, the general behaviour for the controllable states is found from the matrices T_1, T_2.

$$\begin{bmatrix} \omega_{\psi}(t) \\ \psi(t) \end{bmatrix} = \begin{bmatrix} \omega_{\psi}(0) \\ \psi(0) + \omega_{\psi}(0)t \end{bmatrix} + \begin{bmatrix} t & t^2/2 \\ t^2/2 & t^3/6 \end{bmatrix} \begin{bmatrix} a_{s0} \\ a_{s1} \end{bmatrix}. \tag{9.11}$$

The results include the matrix $M_{si}(t)$ which multiplies the initial conditions. For the horizontal guidance here represented with the vector $\mathbf{x}_0(t)$.

$$\begin{bmatrix} \omega_{\psi}(t_f) \\ \psi(t_f) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & t \end{bmatrix} \begin{bmatrix} 1 & 1/2 \\ 1/2 & 1/6 \end{bmatrix} \begin{bmatrix} t & 0 \\ 0 & t^2 \end{bmatrix} \begin{bmatrix} a_{s0} \\ a_{s1} \end{bmatrix} - \begin{bmatrix} 1 & 0 \\ t & 1 \end{bmatrix} \begin{bmatrix} \omega_{\psi}(0) \\ \psi(0) \end{bmatrix} \tag{9.12}$$

$$\boldsymbol{\psi}(t_f) = T_1 A_s T_2 \mathbf{a_s} - M_{si}(t)\boldsymbol{\psi}(0)$$

.

By selecting the time of the manoeuvre, the following equations show how the coefficients are obtained,

$$T_2^{-1} A_s^{-1} T_1^{-1} \left(\boldsymbol{\psi}(t_f) - M_{si} \boldsymbol{\psi}(0) \right) = \mathbf{a_s} \tag{9.13}$$

$$T_2^{-1} A_s^{-1} T_1^{-1} = \begin{bmatrix} 1/t_f & 0 \\ 0 & 1/t_f^{\,2} \end{bmatrix} \begin{bmatrix} -2 & 6 \\ 6 & -12 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1/t_f \end{bmatrix}. \tag{9.14}$$

9.3.1 Simulation Results

The Borea project was born to test landing algorithms. The polynomial guidance implemented in lateral motion and spin motion allows to set all states in an specific time. This property will be important to test other nature of the control strategies and put the quadrotor states in non ideal conditions before to launch landing procedure.

Remark

To get the coefficients is only necessary to define the manoeuvre time and to know the initial conditions; these conditions are obtained from the navigation.

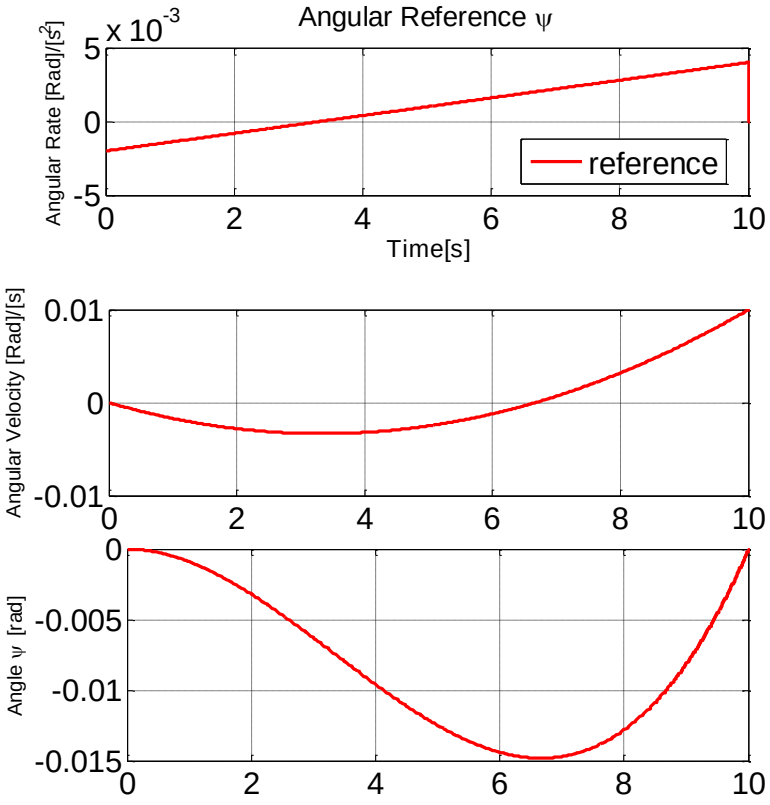

Figure 76. **Spin Reference Generator**

The Figure 76 shows a simple example where a change of angular velocity is required, but the orientation is not modified. The manoeuvre time is defined as 10 s.

10. QUADROTOR NOISE ESTIMATORS

In the quadrotor case states are not available and the sample frequency may be different to control frequency. Each sensor adds noise, and in some cases parasite dynamics, only few variables are measured, for outdoor quadrotors the IMU is the main sensor. For Borea quadrotor the sensor unit was studied and analyzed in [23] and a low level navigation was implemented in [25]. This research uses these works as measurement unit which provides the higher level navigation the state values.

The noise estimators are analyzed in a discrete time, all values are given in meters, to couple with other simulator elements is required to multiply by sample frequency.

Model error is the sole accessible measure of the uncertainty; its current result is defined by the past discrepancies. This model error can be elaborated and accumulated in disturbance states. These residual discrepancies are used to reduce the model error. The model error is always bounded but not zero.

10.1 Vertical Navigation

For the vertical navigation the position model error is defined as e_{lz},

$$e_{lz} = x_{lz} - \hat{x}_{lz} \tag{10.1}$$

To develop the noise estimator, a copy of the embedded model is made. The model error is used to close the loop through the gains l_{z0}, l_{z1}. In order to guarantee the stability, a dynamic filter with state $P(k)$ commanded by the model error is added. The following equation shows the navigation discrete dynamic.

$$\begin{bmatrix} P \\ \hat{x}_{lz} \\ \hat{v}_{lz} \\ D_1 \end{bmatrix}(k+1) = \begin{bmatrix} 1-B_z & 0 & -1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & -l_{z1} & 1 & 1 \\ -m_0 & -l_{z0} & 0 & 1 \end{bmatrix} \begin{bmatrix} P \\ \hat{x}_{lz} \\ \hat{v}_{lz} \\ D_1 \end{bmatrix}(k) + \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} u(k) \tag{10.2}$$

.

The Figure 77 shows the block diagram of the noise estimator,

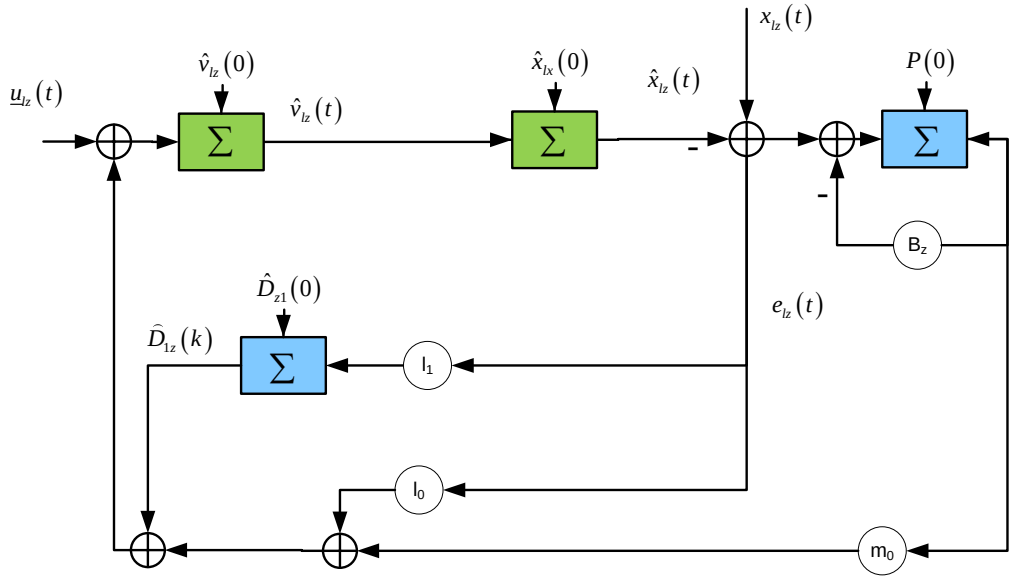

Figure 77. **Vertical EM+Noise Estimator**

To following result is expressed in complementary eigenvalues $\lambda - 1 = \gamma$. To compute the eigenvalues a classical approach is used.

$$\lambda I - A = \begin{bmatrix} \lambda & 0 & 0 & 0 \\ 0 & \lambda & 0 & 0 \\ 0 & 0 & \lambda & 0 \\ 0 & 0 & 0 & \lambda \end{bmatrix} - \begin{bmatrix} 1 & 1 & 0 & 0 \\ -m_0 & 1-p_0 & 1 & 0 \\ -m_1 & 0 & 1 & 1 \\ -m_2 & 0 & 0 & 1 \end{bmatrix} \tag{10.3}$$

$$\det(\lambda I - A) = \gamma^4 + p_0\gamma^3 + m_0\gamma^2 + m_1\gamma + m_2 \quad .$$

At this point, the design of the noise estimator consists in determining the values of l_1, l_2, B_z, m_0. The eigenvalues are set by the designer in order to develop a stable system with a minimum variance.

10.2 Horizontal Navigation

The horizontal navigation uses the embedded model with four order controllable dynamics, a decoupling strategy is implemented to develop the horizontal noise estimator, thanks to

the linear transformation made in the chapter 4 both axes in the horizontal plane follow the same model, therefore the following procedure is valid for them.

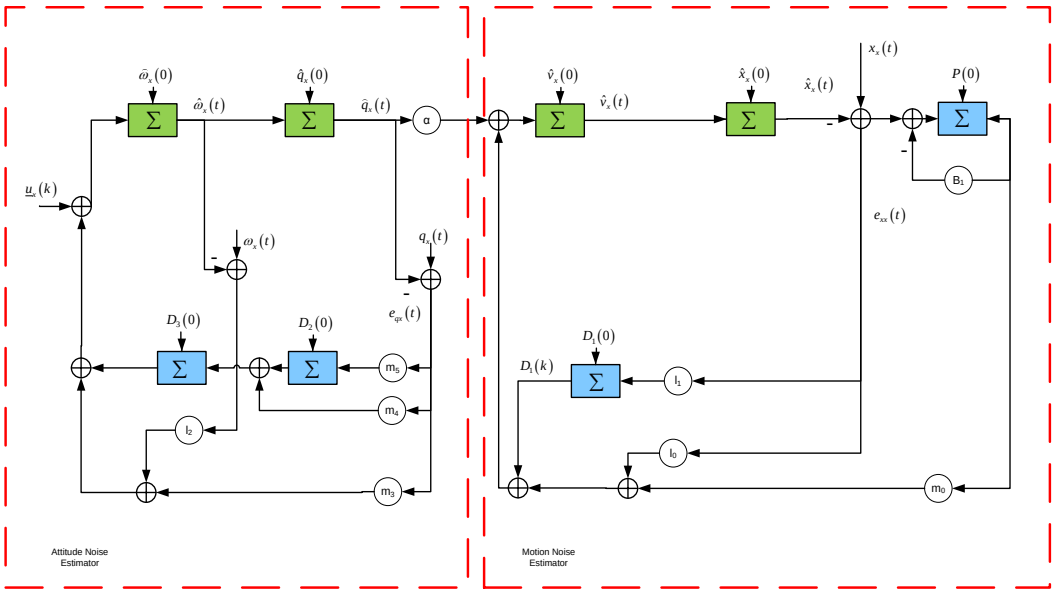

Figure 78. **Horizontal EM + Noise Estimator**

The decupled observer form shown in Figure 78 allows separating the calculations of the navigation. The first observer from the command to acceleration relative to the variables Ω, q is developed assuming that both states can be disturbed, unlike to the velocity and acceleration, since the states Ω, q are obtained from a transformation both of them are susceptible to model errors, the q, ω measures are known through the measure of ω_q and the attitude ϕ, θ, ψ coming from the measure unit and then transform with the feedback linearization. The equation bellow shows the noise estimator dynamics from the command $\dot{\Omega}$ to the state q.

$$\begin{bmatrix} \hat{q}_x \\ \hat{\omega}_x \\ D_{x3} \\ D_{x2} \end{bmatrix}(k+1) = \begin{bmatrix} 1 & 1 & 0 & 0 \\ -m_3 & 1-l_2 & 1 & 0 \\ -m_4 & 0 & 1 & 1 \\ -m_5 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{q}_x \\ \hat{\omega}_x \\ D_{x3} \\ D_{x2} \end{bmatrix}(k) + \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} u_x(k) \qquad (10.4)$$

Here the problem is similar to the previous section, the noise estimator eigenvalues are calculated to guarantee stability with minimum variance. The equation bellow shows the characteristic polynomial in function of the complementary eigenvalues γ.

$$\det\left(\lambda I - A\right) = \gamma^4 + l_2\gamma^3 + m_3\gamma^2 + m_4\gamma + m_5 \tag{10.5}$$

For the model from the acceleration to position, the procedure and the scheme selected are the same of the vertical navigation in section 10.2. The equation bellow shows the noise estimator dynamics from the acceleration to the position.

$$\begin{bmatrix} P \\ \hat{x}_x \\ \hat{v}_x \\ D_{x1} \end{bmatrix}(k+1) = \begin{bmatrix} 1-B_1 & 0 & -1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & -l_1 & 1 & 1 \\ -m_0 & -l_0 & 0 & 1 \end{bmatrix}\begin{bmatrix} P \\ \hat{x}_x \\ \hat{v}_x \\ D_1 \end{bmatrix}(k) + \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}u(k) \tag{10.6}$$

.

It is clear that the results are the same than the previous section. The equation bellow shows the characteristic polynomial in function of the complementary eigenvalues γ.

$$\det\left(\lambda I - A\right) = \gamma^4 + B_1\gamma^3 + l_1\gamma^2 + \left(l_0 - m_0 + B_1 l_1\right)\gamma + B_1 l_0 \tag{10.7}$$

10.3 Spin Navigation

The spin navigation uses the same scheme and strategy of the vertical noise estimator, there are not expected trajectories in the spin, the objective is to guarantee the velocity equal to zero, the algorithm presented in this research brings the spin angle to zero.

11. QUADROTOR CONTROL LAW

11.1 Vertical Control Law

The CoM control is made in the inertial frame of reference, the controllable states are included to develop a state feedback, thus the command $u_{lz}(k)$ from this point is divided in two components, the reference command and the feedback state command

$$u_{lz}(k) = \underline{u}_{lz}(k) + \breve{u}_{lz}. \tag{11.1}$$

The feedback command $\breve{u}_{lz}$ is generated from a linear combination of the tracking errors $\breve{e}_{lxz}(k), \breve{e}_{lvz}(k)$, these errors are multiplied by the feedback gains k_p, k_v. The new state model is

$$\begin{bmatrix} x_{lz} \\ v_{lz} \end{bmatrix}(k+1) = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} x_{lz} \\ v_{lz} \end{bmatrix}(k) + \begin{bmatrix} 0 \\ 1 \end{bmatrix}\left(\underline{u}_{lz}(k) + \breve{u}_{lz}\right). \tag{11.2}$$

The Figure 79 shows the control law diagram block, the feedback is closed with the estimated variables, the tracking error with the real states is unavailable for any control strategy.

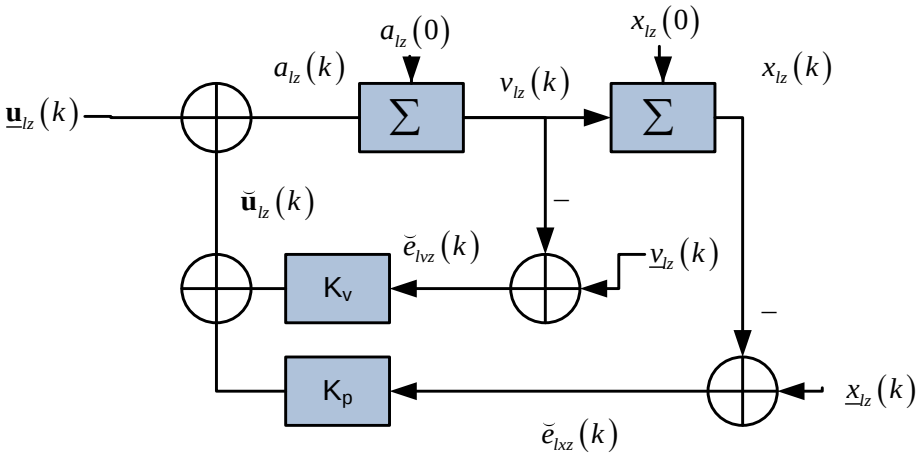

Figure 79. **Vertical Control Law**

As the classical approach the dynamic of the system is modified by the feedback action, the equation bellow shows the state representation of the feedback system

$$\begin{bmatrix} x_{lz} \\ v_{lz} \end{bmatrix}(k+1) = \begin{bmatrix} 1 & 1 \\ -k_p & 1-k_v \end{bmatrix}\begin{bmatrix} x_{lz} \\ v_{lz} \end{bmatrix}(k) + \begin{bmatrix} 0 \\ 1 \end{bmatrix}u(k) + \begin{bmatrix} \underline{x}_{lz} \\ \underline{v}_{lz} \end{bmatrix}(k) \quad (11.3)$$

These procedures are shown to provide the reader with the simple procedures that are required to get the feedback gains and therefore the desired eigenvalues. The calculation of the characteristic polynomial

$$\det\left(\begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix} - \begin{bmatrix} 1 & 1 \\ -k_p & 1-k_v \end{bmatrix}\right) = \det\left(\begin{bmatrix} \lambda-1 & -1 \\ k_p & \lambda-1+k_v \end{bmatrix}\right) \quad (11.4)$$

As in previous chapters the procedures are simplified when they are expressed in complementary eigenvalues $\lambda - 1 = \gamma$ .The characteristic polynomial is shown in the following equation.

$$\det\left(\begin{bmatrix} \gamma & -1 \\ k_p & \gamma + k_v \end{bmatrix}\right) = \gamma^2 + \gamma k_v + k_p \quad (11.5)$$

The eigenvalues are selected to guarantee stability and minimum variance.

11.2 Horizontal Control Law

The lateral control law follows the same strategy of vertical control law, therefore four tracking errors are presented $\breve{e}_{\Omega x}, \breve{e}_{qx}, \breve{e}_{vx}, \breve{e}_{xx}$ to define the command action $\breve{u}_x$, Figure 80 shows the block diagram of the horizontal control law.

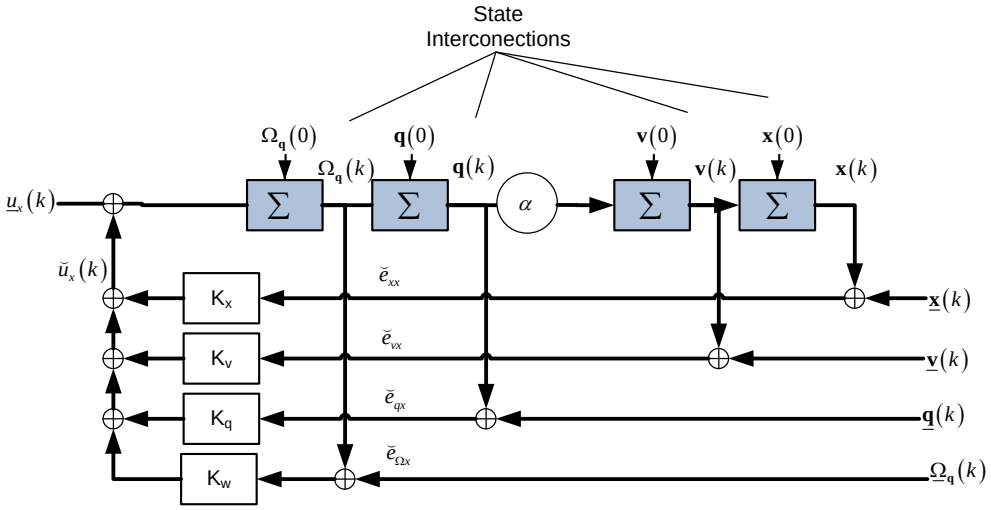

Figure 80. **Horizontal Control law**

The model used to develop the control law includes the interconnection states generated by the discrete approximation, this bring the digital model closer to the real model. The following equation shows the discrete dynamic model considered to the control law, the vector $\mathbf{K}_h$ defines the dynamics of the horizontal position system.

$$\mathbf{K}_h = \begin{bmatrix} k_p & k_v & k_q & k_{\omega q} \end{bmatrix} \tag{11.6}$$

$$\mathbf{x}(k+1) = (A + B\mathbf{K}_h)\mathbf{x}(k) + B\underline{u}(k) \tag{11.7}$$

$$(A + B\mathbf{K}_h) = \begin{bmatrix} 1 & 1 & \frac{\alpha}{2} & \frac{\alpha}{6} \\ 0 & 1 & \alpha & \frac{\alpha}{2} \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} \frac{\alpha}{24} \\ \frac{\alpha}{6} \\ \frac{1}{2} \\ 1 \end{bmatrix} \begin{bmatrix} k_p & k_v & k_q & k_{\omega q} \end{bmatrix} \tag{11.8}$$

.

The equation bellow shows the characteristic polynomial of the horizontal control law,

$$\begin{aligned} \det(A + B\mathbf{K}_h) &= \gamma^4 + \left(\frac{\alpha k_p}{24} + \frac{\alpha k_v}{6} + \frac{k_q}{2} + k_w \right)\gamma^3 \\ &+ \left(\frac{7\alpha k_p}{12} + \alpha k_v + k_q \right)\gamma^2 + \left(\frac{3\alpha k_p}{2} + \alpha k_v \right)\gamma + \alpha k_p \end{aligned} \tag{11.9}$$

.

To calculate the value of the vector $\mathbf{K}_h$ is necessary to know the α value, but this value can change at each time step, these constants are adapted in order to validate the lineal model.

11.3 Spin Control

The spin control implemented is an angle control. The target is always zero, it may be different due to developed of the guidance and navigation, following phases of the Borea project can be included aggressive manoeuvre, obstacles avoidance or optimal consume, in these cases the spin control can be useful.

12. SIMILITUDE CASE

12.1 How to emulate landing

The dynamics and kinematics of a spacecraft in the propulsive phase have several features in common with a quadrocopter vehicle the chapters 4 and 7 show this similitude, for the spacecraft the propulsion assembly has three characteristics; a near axial force is applied, Horizontal displacement is driven by axis tilt and Axis tilt is commanded by torques (Figure 81 left). On the other hand for quad-rotor propulsion, thrust is axial, horizontal displacement is driven by axis tilt, and axis tilt is commanded by torques (Figure 81 right), as in the spacecraft case.

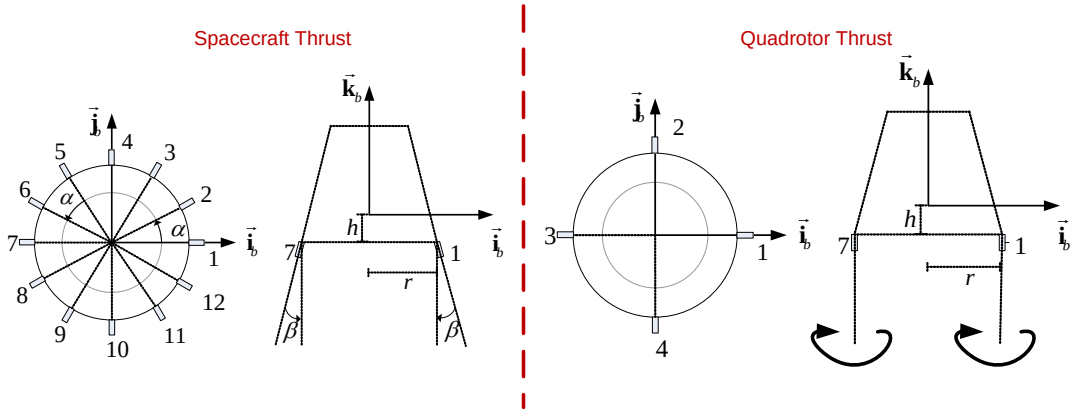

Figure 81. **Spacecraft-quadrotor geometry comparison.**

For sensors the Table 8 shows the comparison between both measurement units, from emulating radar altimeter and velocimeter; both can be emulated by a GPS receiver but reliably only in outdoor tests. Indoor tests should require camera. Radar altimeters are massive. Altimeter may be also emulated by a barometric altimeter. Ultrasonic range sensors are used at touch-down. Initial alignment must be provided by some attitude sensor either magnetometers or external markers, or the accelerometer themselves (on-ground).

Table 8. Spacecraft – quadrotor comparison

No		BOREA	Landing vehicle

0	IMU (accelerometer and gyro) 3D	X	X
1	Radar altimeter 1D		X
2	Proximity sensor (ultrasound)	X	
3	Radar velocimeter 3D		X
4	GPS (position)	X	
5	Magnetometer (attitude) 2D	X	
6	Near axial thrusters		X
7	4 axial propellers and electric motors	X	

12.2 Similitude Test

In space applications the initial conditions cannot be set, the initial vertical and horizontal velocities and attitude are defined after the atmospheric entry therefore, the similitude test aims validate the landing algorithms. To achieve this is mandatory to be able to lead the quadrotor to different conditions of velocity and attitude before the landing. For that reason the Similitude test is divided into three phases: the first one is the lifting, which leads the quadrotor at a certain height, the second one is the initialization; in this phase the velocity conditions and initial attitude are selected, and finally the landing phase.

Guidance aims to provide the reference trajectory for the vertical and horizontal models. Guidance is adaptive in the sense that is recomputed at change of the reference (maneuver initial conditions are taken account).

Navigation in section 10.1 has been done parametrically, the algorithms that are used in this work are not limited by the scale run, the dynamic input is the force applied by the propellers studied and modeled in section 7.3.1, i.e. the thrusters force is exchanged for propeller force.

The similitude tests are made with the three main phases mentioned before, the initial conditions are define in order to carry the quadrotor a non ideal conditions before to apply the landing algorithms as in the real conditions in power descent phase to this aims the vertical and horizontal guidance in chapter 9 is applied. The dissertation shows the results of different tests that introduce variations in; vertical velocity, 3 axis angular positions and 3 axis angular velocities.

12.3 Simulation Results

During the initial conditions setting the position cannot be defined, i.e, the initial position is not controlled, after setting the initial conditions, the quadrotor has as target a horizontal displacement of 20 meters from its current position.

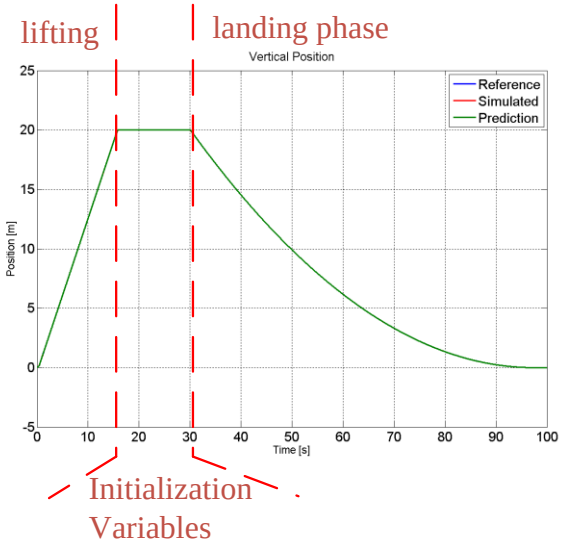

Figure 82. **Vertical position - Similitude Phases**

Figure 83 shows a comparison between the horizontal position of four tests. According to this result, the guidance algorithm is updated and dependent on initial conditions. The path

is clearly different due to variations that are set in the initialization variables phase (Figure 82).

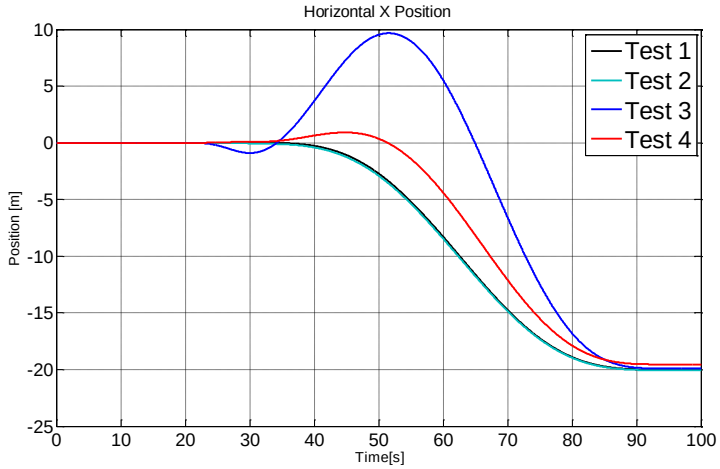

Figure 83. **Horizontal Position – Similitude Test**

The Figure 84 shows the horizontal velocity, in all cases the desired horizontal displacement is achieved and the horizontal velocity is brought to zero implying coupled model for the orientation and angular velocity are also carried at zero value.

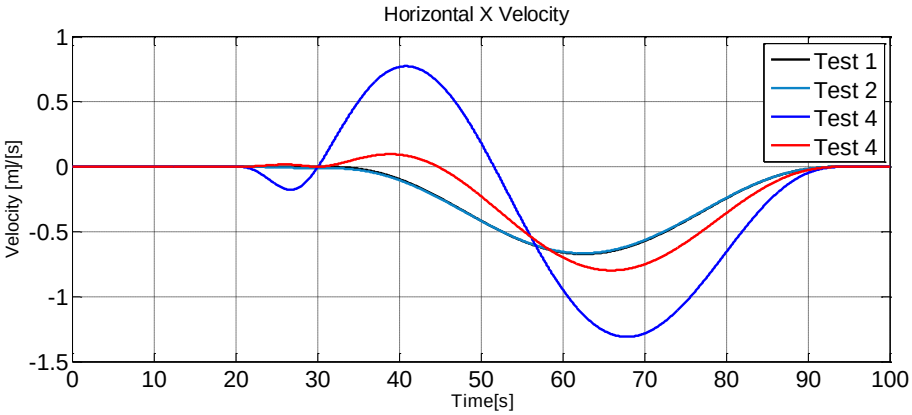

Figure 84. **Horizontal velocity**

The Figure 85 shows a typical 3D position on a similitude test, the quadrotor path spends 97 seconds maneuver.

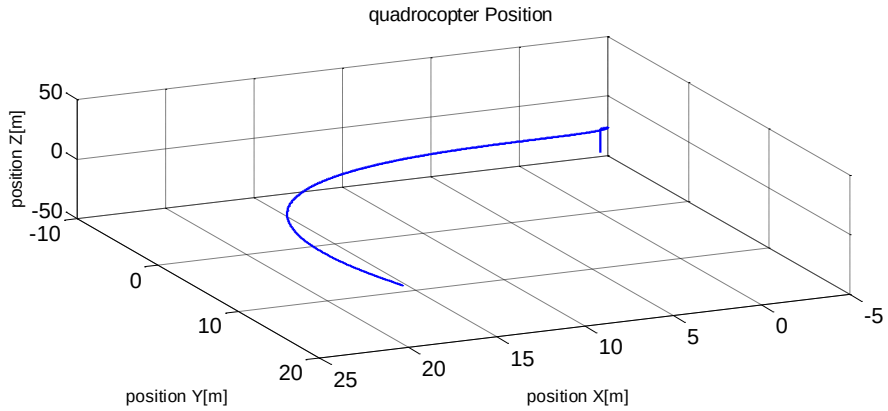

Figure 85. **Quadrocopter Position 3D**

13. CONCLUSIONS

Algorithms of guidance and control for pinpoint planetary landing on propulsive phase were studied and integrated with a piloting strategy. The piloting strategy to avoid hazards based on computer vision that considers the bounds of a single camera impose was developed, a generic geometric description of the propulsive system was studied and an optimal dispatching strategy was performed.

This research allowed to develop GNC based on EMC method for quadrotor vehicles. Embedded models, noise estimators and control laws were developed and simulated for vertical and horizontal position. A scientific quadrotor was built from scratch in the framework of Borea quadrotor project (Figure 86). A planetary similitude case was studied and simulated.

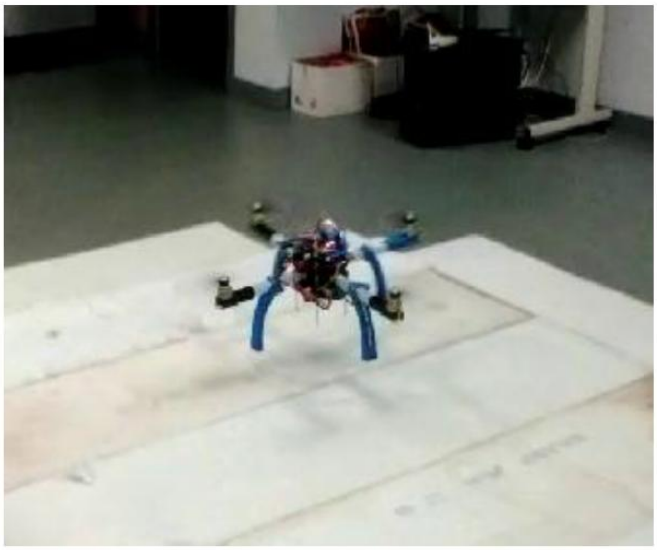

Figure 86. **Borea Quadrotor**

14. REFERENCES

[1] E. Canuto and L. D. Prieto, “Embedded model control: principles and applications. part I,” in *2007 IEEE Conference on Emerging Technologies & Factory Automation (EFTA 2007)*, 2007, pp. 45–52.

[2] E. Canuto and L. D. Prieto, “Embedded model control: principles and applications. part II,” in *2007 IEEE Conference on Emerging Technologies & Factory Automation (EFTA 2007)*, 2007, pp. 53–60.

[3] E. Canuto, W. Acuna-Bravo, A. Molano-Jimenez, J. Ospina, and C. Perez-Montenegro, “Embedded Model Control urges disturbance modelling and rejection.” pp. 6267–6273, 2011.

[4] L. Massotti, E. Canuto, and P. Silvestrin, “Embedded Model Control Application to Drag-Free and Satellite-to-Satellite Tracking,” in *2006 14th Mediterranean Conference on Control and Automation*, 2006, pp. 1–6.

[5] E. Canuto, W. Acuna-Bravo, A. M. Jimenez, and C. P. Montenegro, “The error loop and robust closed-loop stability,” in *2012 IEEE International Conference on Mechatronics and Automation*, 2012, pp. 1115–1120.

[6] E. Canuto, “Embedded Model Control: outline of the theory.,” *ISA Trans.*, vol. 46, no. 3, pp. 363–77, Jun. 2007.

[7] A. Jimenez Molano, “Embedded Model Control for Mars Terminal Descent Phase,” Politecnico di Torino, 2011.

[8] M. S. Martin, G. F. Mendeck, P. B. Brugarolas, G. Singh, and F. Serricchio, “reconstructed flight performance of the mars science laboratory guidance, navigation system for entry, descent, and landing,” in *Anual Guidance Navigation and Control Conference*, 2014, p. 30.

[9] R. Prakash, P. D. Burkhart, A. Chen, K. A. Comeaux, C. S. Guernsey, D. M. Kipp, L. V. Lorenzoni, G. F. Mendeck, R. W. Powell, T. P. Rivellini, A. M. S. Martin, S. W. Sell, A. D. Steltzner, and D. W. Way, “Mars Science Laboratory Entry, Descent, and Landing System Overview,” in *2008 IEEE Aerospace Conference*, 2008, pp. 1–18.

[10] R. D. Braun and R. M. Manning, “Mars Exploration Entry, Descent and Landing Challenges,” in *2006 IEEE Aerospace Conference*, 2006, pp. 1–18.

[11] E. Canuto, A. Molano-Jimenez, W. Acuna-Bravo, C. Perez-Montenegro, and S. Malan, “Planetary landing: Modelling and control of the propulsive descent.” pp. 7309–7316, 2012.

[12] E. Canuto, A. Jimenez Molano, C. Perez Montenegro, S. Malan, P. Martella, A. Molano-Jimenez, and C. Perez-Montenegro, “Planetary landing: Modelling and control of the propulsion descent,” *J. Univ. Sci. Technol. China*, vol. 43, pp. 1–14, 2013.

[13] S. Ploen, B. Acikmese, and A. Wolf, “A Comparison of Powered Descent Guidance Laws for Mars Pinpoint Landing,” in *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, 2006.

[14] J. M. Carson, B. Acikmese, L. Blackmore, and A. A. Wolf, “Capabilities of convex Powered-Descent Guidance algorithms for pinpoint and precision landing,” in *2011 Aerospace Conference*, 2011, pp. 1–8.

[15] T. Tomic, K. Schmid, P. Lutz, A. Domel, M. Kassecker, E. Mair, I. Grixa, F. Ruess, M. Suppa, and D. Burschka, “Toward a Fully Autonomous UAV: Research Platform for Indoor and Outdoor Urban Search and Rescue,” *IEEE Robot. Autom. Mag.*, vol. 19, no. 3, pp. 46–56, Sep. 2012.

[16] A. B. Aç\ikmese and S. R. Ploen, “A powered descent guidance algorithm for Mars pinpoint landing,” *Tc*, vol. 1000, p. 1, 2005.

[17] L. Blackmore, B. Acikmese, and D. P. Scharf, “Minimum-Landing-Error Powered-Descent Guidance for Mars Landing Using Convex Optimization,” *J. Guid. Control. Dyn.*, vol. 33, no. 4, pp. 1161–1171, Jul. 2010.

[18] L. Piergiorgio, N. Nicoletta, M. Corrado, T. Antonio, Z. Luca, and O. Francesca, “A Vision-Based Navigation Facility for Planetary Entry Descent Landing,” in *Computer Vision – ECCV 2012. Workshops and Demonstrations*, 2012, p. pp 546–555.

[19] B. Parreira, J. F. Vasconcelos, R. Oliveira, A. Caramagno, P. Motrena, and J. Dinis, “Performance Assessment Of Vision Based Hazard Avoidance During Lunar And Martian Landing,” no. 1.

[20] E. S. Crane and S. M. Rock, “Guidance augmentation for reducing uncertainty in vision-based hazard mapping during lunar landing.” pp. 1–12, 2013.

[21] C. Perez-Montenegro, M. Lotufo, and E. Canuto, “Control Architecture and Simulation of the Borea Quadrotor,” in *Research, Education and Development of Unmanned Aerial Systems*, 2013, vol. 2, no. 1, pp. 168–173.

[22] E. Canuto and C. Perez-Montenegro, “Modelling And Control Of A Small Quadrotor For Testing Propulsive Planetary Landing Guidance, Navigation And Control,” 2012, vol. IAC-12.

[23] S. Lopez-Carmona, “Hadware Implementation of Measurement Unit for Borea Quadricopter,” Politecnico di torino, 2012.

[24] C. Circa, “Modelistica e simulazione del sistema di propulsuione del quadrirotore Borea,” Politecnico di Torino, 2012.

[25] S. Mosca, “Modellistica e Simulazione Della Navigazione Di un Quadricottero,” Politecnico di Torino, 2012.

[26] C. Perez-Montenegro and E. Canuto, “Borea Official Web Site,” 2014. [Online]. Available: http://spautomatics.polito.it/S&PA/borea.php.

[27] D. W. Way, R. W. Powell, A. Chen, A. D. Steltzner, A. M. S. Martin, P. D. Burkhart, and G. F. Mendeck, “Mars Science Laboratory: Entry, Descent, and Landing System Performance,” in *2007 IEEE Aerospace Conference*, 2007, pp. 1–19.

[28] M. J. Grant, B. A. Steinfeldt, R. D. Braun, and G. H. Barton, “Smart Divert: A New Mars Robotic Entry, Descent, and Landing Architecture,” *J. Spacecr. Rockets*, vol. 47, no. 3, pp. 385–393, May 2010.

[29] R. D. Braun and R. M. Manning, “Mars Exploration Entry, Descent, and Landing Challenges,” *J. Spacecr. Rockets*, vol. 44, no. 2, pp. 310–323, Mar. 2007.

[30] S. J. CITRON, S. E. DUNIN, and H. F. MEISSINGER, “A terminal guidance technique for lunar landing,” *AIAA J.*, vol. 2, no. 3, pp. 503–509, Mar. 1964.

[31] C. R. McInnes, “Path shaping guidance for terminal lunar descent,” *Acta Astronaut.*, vol. 36, no. 7, pp. 367–377, 1995.

[32] C. R. McInnes, “Direct Adaptive Control for Gravity-Turn Descent,” *J. Guid. Control. Dyn.*, vol. 22, no. 2, pp. 373–375, Mar. 1999.

[33] H. Hablani, "Interplanetary spacecraft controllers using thrusters," in *Guidance, Navigation, and Control Conference*, 1997.

[34] P. Martella, M. Buonocore, E. Canuto, A. Molano-Jimenez, R. Drai, and L. Lorenzoni, "Design and Verification of the GNC for the European ExoMars EDL Demonstrator," in *AIAA Guidance, Navigation, and Control Conference*, 2011.

[35] J. de Lafontaine, D. Neveu, and K. Lebel, "Autonomous Planetary Landing with Obstacle Avoidance: The Quartic Guidance Revisited," in *Advances in Astronautical Sciences*, 2004, pp. 2717–2742.

[36] E. C. Wong, G. Singh, and J. P. Masciarelli, "Guidance and Control Design for Hazard Avoidance and Safe Landing on Mars," *J. Spacecr. Rockets*, vol. 43, no. 2, pp. 378–384, Mar. 2006.

[37] G. Singh, A. M. SanMartin, and E. C. Wong, "Guidance and Control Design for Powered Descent and Landing on Mars," in *2007 IEEE Aerospace Conference*, 2007, pp. 1–8.

[38] F. Najson and K. D. Mease, "Computationally Inexpensive Guidance Algorithm for Fuel-Efficient Terminal Descent," *J. Guid. Control. Dyn.*, vol. 29, no. 4, pp. 955–964, Jul. 2006.

[39] R. Sostaric and J. Rea, "Powered Descent Guidance Methods For The Moon and Mars," in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2005.

[40] J. de Lafontaine, D. Neveu, K. Lebel, and J. De Lafontaine, "Autonomous planetary landing using a LIDAR sensor: the closed-loop system," in *Guidance, Navigation and Control Systems*, 2006, vol. 606, p. 3.

[41] J. A. Guerrero, P. Castillo, Y. Challal, and J. A. G. P. Castillo, "Trajectory Tracking for a Group of Mini Rotorcraft Flying in Formation," in *World Congress*, 2011, vol. 18, no. 1, pp. 6331–6336.

[42] J.-J. E. Slotine, W. Li, and others, *Applied nonlinear control*, vol. 199, no. 1. Prentice hall New Jersey, 1991.

[43] E. Canuto, W. Acuna-Bravo, A. Molano-Jimenez, and C. Perez Montenegro, "Embedded model control calls for disturbance modeling and rejection," *ISA Trans.*, vol. 51, no. 5, pp. 584–595, 2012.

[44] P. C. Hughes, *Spacecraft attitude dynamics*. Dover Publications, 2004.

[45] M. H. Kaplan, *Modern spacecraft dynamics and control*, vol. 1. New York,: John Wiley and Sons, Inc., 1976.

[46] B. Acikmese and S. R. Ploen, "Convex Programming Approach to Powered Descent Guidance for Mars Landing," *J. Guid. Control. Dyn.*, vol. 30, no. 5, pp. 1353–1366, Sep. 2007.

[47] D. G. Luenberger, *Optimization by Vector Space Methods*. 1968.

[48] C. Perez-Montenegro, M. Lotufo, L. Colangelo, and E. Canuto, "Saturation Managing for the Propulsion System and Optimal Dispatching Proposal for multi-rotor UAV," in *2014 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, 2014, p. 6.

[49] K. Zhou, J. C. Doyle, and G. K., *Robust and optimal control, Prentice Hall*. Upper Saddle River, NJ, 1995.

[50] K. R. Muske and T. A. Badgwell, "Disturbance modeling for offset-free linear model predictive control," *J. Process Control*, vol. 12, no. 5, pp. 617–632, Aug. 2002.

[51] C. D. Johnson, "Disturbance-Accommodating Control; An Overview," 1986, pp. 526–536.

[52] R. Kalman, "On the general theory of control systems," *IRE Trans. Autom. Control*, vol. 4, no. 3, p. 110, 1959.

[53] E. Canuto, A. Molano, and L. Massotti, "Drag-Free Control of the GOCE Satellite: Noise and Observer Design," *IEEE Trans. Control Syst. Technol.*, vol. 18, no. 2, pp. 501–509, Mar. 2010.

[54] J. M. Carson III, B. Acikmese, L. Blackmore, and J. M. C. Iii, "Lossless convexification of Powered-Descent Guidance with non-convex thrust bound and pointing constraints," *Am. Control Conf. (ACC), 2011*, pp. 2651–2656, 2011.

[55] a. Johnson and L. Matthies, "MER-DIMES: a planetary landing application of computer vision," *2005 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 806–813 vol. 1, 2005.

[56] Y. Cheng, "Real-time surface slope estimation by homography alignment for spacecraft safe landing," *2010 IEEE Int. Conf. Robot. Autom.*, pp. 2280–2286, May 2010.

[57] L. Matthies, A. Huertas, Y. Cheng, and A. Johnson, "Stereo vision and shadow analysis for landing hazard detection," *2008 IEEE Int. Conf. Robot. Autom.*, pp. 2735–2742, May 2008.

[58] A. E. Johnson, A. Huertas, R. A. Werner, and J. F. Montgomery, "Analysis of On-Board Hazard Detection and Avoidance for Safe Lunar Landing," in *2008 IEEE Aerospace Conference*, 2008, pp. 1–9.

[59] A. Huertas and R. Madison, "Passive Imaging Based Multicue Hazard Detection for Spacecraft Safe Landing," in *2006 IEEE Aerospace Conference*, 2006, pp. 1–14.

[60] Y. Cheng, A. Johnson, L. Mattheis, and A. Wolf, "Passive imaging based hazard avoidance for spacecraft safe landing," 2001.

[61] S. R. Ploen, H. Seraji, and C. E. Kinney, "Determination of Spacecraft Landing Footprint for Safe Planetary Landing," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 45, no. 1, pp. 3–16, Jan. 2009.

[62] D. Garcia-Castellanos and U. Lombardo, "Poles of inaccessibility: A calculation algorithm for the remotest places on earth," *Scottish Geogr. J.*, vol. 123, no. 3, pp. 227–233, Sep. 2007.

[63] M. Y. Amir and V. Abbass, "Modeling of Quadrotor Helicopter Dynamics," in *2008 International Conference on Smart Manufacturing Application*, 2008, pp. 100–105.

[64] L. Tan, L. Lu, and G. Jin, "Attitude stabilization control of a quadrotor helicopter using integral backstepping," *Int. Conf. Autom. Control Artif. Intell. (ACAI 2012)*, pp. 573–577, 2012.

[65] A. C. Satici, H. Poonawala, and M. W. Spong, "Robust Optimal Control of Quadrotor UAVs," *IEEE Access*, vol. 1, pp. 79–93, 2013.

[66] S. Grzonka, G. Grisetti, and W. Burgard, "A Fully Autonomous Indoor Quadrotor," *IEEE Trans. Robot.*, vol. 28, no. 1, pp. 90–100, Feb. 2012.

[67] Z. Zuo, "Trajectory tracking control design with command-filtered compensation for a quadrotor," *IET Control Theory Appl.*, vol. 4, no. 11, pp. 2343–2355, Nov. 2010.

[68] a. Tzes, G. Nikolakopoulos, and K. Alexis, "Model predictive quadrotor control: attitude, altitude and position experimental studies," *IET Control Theory Appl.*, vol. 6, no. 12, pp. 1812–1827, Aug. 2012.

[69] R. Mahony, V. Kumar, and P. Corke, “Multirotor Aerial Vehicles: Modeling, Estimation, and Control of Quadrotor,” *IEEE Robot. Autom. Mag.*, vol. 19, no. 3, pp. 20–32, Sep. 2012.

[70] S. Bouabdallah, M. Becker, and R. Siegwart, “Autonomous miniature flying robots: coming soon! - Research, Development, and Results,” *IEEE Robot. Autom. Mag.*, vol. 14, no. 3, pp. 88–98, Sep. 2007.

[71] L. García-Delgado, a. Dzul, V. Santibáñez, and M. Llama, “Quad-rotors formation based on potential functions with obstacle avoidance,” *IET Control Theory Appl.*, vol. 6, no. 12, pp. 1787–1802, Aug. 2012.

[72] R. Yanushevsky, *Guidance of Unmanned Aerial Vehicles*. Taylor & Francis, 2011.

[73] W. Zang, J. Lin, Y. Wang, and H. Tao, “Investigating small-scale water pollution with UAV Remote Sensing Technology.” pp. 1–4, 2012.

[74] P. Gao, L. Yan, H. Zhao, and S. Lu, “Study on shooting control algorithm of remote sensing control system for UAV,” *2007 IEEE Int. Geosci. Remote Sens. Symp.*, pp. 623–625, 2007.

[75] G. Cook, *Mobile Robots: Navigation, Control and Remote Sensing*. 2011.

[76] A. Restas, “Forest Fire Management Supporting by UAV Based Air Reconnaissance Results of Szendro Fire Department, Hungary,” in *2006 First International Symposium on Environment Identities and Mediterranean Area*, 2006, pp. 73–77.

[77] F. Hoffmann, N. Goddemeier, and T. Bertram, “Attitude estimation and control of a quadrocopter,” in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, pp. 1072–1077.

[78] S. Formentin and M. Lovera, “Flatness-based control of a quadrotor helicopter via feedforward linearization,” in *IEEE Conference on Decision and Control and European Control Conference*, 2011, pp. 6171–6176.

[79] J. Tan, C. Y. Dat, T. J. Quan, and A. E. Ling, “Design of a Easy to Fly, Low-Cost Generic Unmanned Aerial Vehicle (UAV) for Civilian Aerial-Imaging Application,” no. October, pp. 283–288, 2012.

[80] J. Tan, A. E. Ling, T. J. Quan, and C. Y. Dat, “Generic Unmanned Aerial Vehicle (UAV) for civilian application,” no. October, pp. 289–294, 2012.

[81] E. Semsch, M. Jakob, D. Pavlicek, and M. Pechoucek, “Autonomous UAV Surveillance in Complex Urban Environments,” *2009 IEEE/WIC/ACM Int. Jt. Conf. Web Intell. Intell. Agent Technol.*, pp. 82–85, 2009.

[82] A. Jaimes, S. Kota, and J. Gomez, “An approach to surveillance an area using swarm of fixed wing and quad-rotor unmanned aerial vehicles UAV(s),” in *2008 IEEE International Conference on System of Systems Engineering*, 2008, pp. 1–6.

[83] J. M. Sullivan, “Revolution or evolution? the rise of the UAVs,” *Proceedings. 2005 Int. Symp. Technol. Soc. 2005. Weapons Wires Prev. Saf. a Time Fear. ISTAS 2005.*, pp. 95–102, 2005.

[84] R. Schneiderman, “Unmanned Drones are Flying High in the Military/Aerospace Sector [Special Reports],” *IEEE Signal Process. Mag.*, vol. 29, no. 1, pp. 8–11, Jan. 2012.

[85] a. Azzam, “Quad rotor arial robot dynamic modeling and configuration stabilization,” in *2010 2nd International Asia Conference on Informatics in Control, Automation and Robotics (CAR 2010)*, 2010, pp. 438–444.

[86] A. Tayebi and S. McGilvray, “Attitude stabilization of a four-rotor aerial robot,” in *2004 43rd IEEE Conference on Decision and Control (CDC) (IEEE Cat. No.04CH37601)*, 2004, vol. 2, pp. 1216–1221 Vol.2.

[87] J. Seddon, *Basic Helicopter Aerodynamics*. 1990.

[88] J. Roskam and C.-T. E. Lan, *Airplane Aerodynamics and Performance*. 1997.

[89] V. Khorani and A. Mohammad Shahri, “The true role of accelerometers in quadrotor’s Inertial Navigation System,” in *2013 First RSI/ISM International Conference on Robotics and Mechatronics (ICRoM)*, 2013, pp. 474–480.

[90] P. Martin and E. Salaun, “The true role of accelerometer feedback in quadrotor control,” in *2010 IEEE International Conference on Robotics and Automation*, 2010, pp. 1623–1629.

[91] S. Marchiafava, *Quaternionic Structures in Mathematics and Physics*. Rome, Italy, 2001.

[92] J. B. Kuipers, *Quaternions and Rotation Sequences: A Primer with Applications to Orbits, Aerospace, and Virtual Reality*. Princeton University Press, 2002, p. 371.

[93] S. J. Dodds, “Bang-bang control law for single-input time-invariant plant,” vol. 128, no. 5. pp. 227–232, 1981.